Model-Based 3D Face Capture with Shape-from-Silhouettes

Baback Moghaddam [†] Jinho Lee [‡] Hanspeter Pfister [†] Raghu Machiraju [‡]

[†] Mitsubishi Electric Research Laboratory
201 Broadway, Cambridge MA 02139 USA
{baback,pfister}@merl.com

[‡] The Ohio State University
2015 Neil Avenue, Columbus OH 43210 USA
{leeji,raghu}@cis.ohio-state.edu

Abstract

We present a method for 3D face acquisition using a set or sequence of 2D binary silhouettes. Since silhouette images depend only on the shape and pose of an object, they are immune to lighting and/or texture variations (unlike feature or texture-based shape-from-correspondence). Our prior 3D face model is a linear combination of "eigenheads" obtained by applying PCA to a training set of laser-scanned 3D faces. These shape coefficients are the parameters for a near-automatic system for capturing the 3D shape as well as the 2D texture-map of a novel input face. Specifically, we use back-projection and a boundary-weighted XOR-based cost function for binary silhouette matching, coupled with a probabilistic "downhill-simplex" optimization for shape estimation and refinement. Experiments with a multi-camera rig as well as monocular video sequences demonstrate the advantages of our 3D modeling framework and ultimately, its utility for robust face recognition with built-in invariance to pose and illumination.

1. Introduction

Recently it has become clear that the two most critical factors limiting the performance of automatic face recognition systems are pose and illumination. Therefore, it follows logically that the best and most complete solution to this problem is to acquire/analyze/match a full 3D model of the face as represented, for example, by a 3D shape-mesh plus a 2D texture-map. While 2D view-based [11] and other appearance-based approaches definitely have merit, they suffer from a fundamentally limited representation (a collection of 2D appearance subspaces). We believe that an intrinsic 3D model is the only way to properly tackle the complications that arise due to pose and illumination (and perhaps expression as well) and that they will pave the way for the next generation of robust face-recognition systems.

Traditionally, an image or span of images in time and/or space has been the most common means to convey the information about shape and/or motion of objects in the real world. A single image has natural limitations in revealing 3D information of the objects given the reduction of one dimension. However, this limitation can be overcome by a span of images in time (a video sequence of objects in motion) or a span of images in space (images captured from different viewpoints) or a combination of the two. The traditional approach is of course that of *Structure-from-Motion* (SFM) [5, 20, 17]. It deals with the problem of recovering 3D points on a rigid object from 2D correspondences of points across images. SFM is a direct method to obtain the 3D points which are often not so dense or accurate, thus a post-processing phase is required.

One of the strongest clues for the 3D information contained in a 2D image is the outline of an object in the image. *Shape-from-Silhouette* (SFS) techniques have been used to reconstruct 3D shapes from multiple silhouette images of an object without previous knowledge of the object to be reconstructed [9, 16]. The reconstructed 3D shape is called a visual hull, which is a maximal approximation of the object consistent with the object's silhouettes. The accuracy of this approximate visual hull depends on the number and location of the cameras used to generate the input silhouettes. In general, a complex object such as the human face does not yield a good shape when approximated by a visual hull using a small number of cameras. Moreover, human faces possess concavities (e.g. eye sockets and philtrum) which are impossible to reconstruct even in an exact visual hull due to its inherent limitation. An example of this, taken from our dataset is shown in Figure 1 which shows a 3D shape of a subject alongside the approximated visual hull obtained (synthetically) via 50 viewpoints. However, using knowledge of the object to be reconstructed, silhouette information can still be exploited as an important constraint for the exact shape of the object.

In this paper, we present two systems that recover 3D shape of a human face (and consequently its 2D texture-map as well) from a sequence of silhouette images using an underlying model-based 3D face shape statistical prior. A single monocular video stream as well as a multiple-camera rig are used as input sources (in both synthetic and real experiments) to demonstrate the effectiveness of our model-based

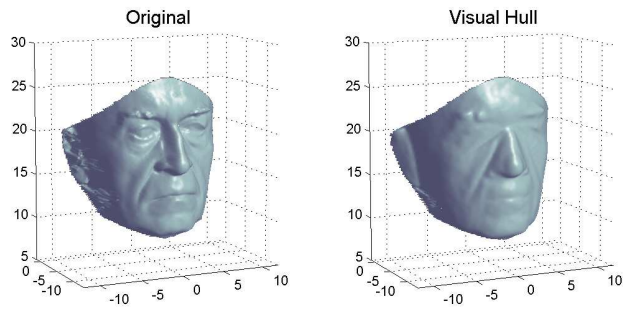

Figure 1: Original laser-scanned face vs. its own Visual Hull, obtained using 50 viewpoints (synthetically rendered).

shape-from-silhouette matching approach. In Section 3, we first describe the "eigenhead" face shape model. Section 4 formulates the inverse problem of reconstructing a 3D face from its silhouettes, using a novel "correspondence-free" boundary-weighted XOR-based cost function for direct 2D matching and the subsequent 2D face "texture-lifting". Section 5 presents two (roughly equivalent) operational scenarios: using a single monocular video sequence vs. a multiple-camera rig "snapshots". Experimental results with both synthetic and real data (in both scenarios) are presented in Section 6. Finally, Section 7 has a brief summary and discussion of our results and future plans.

2. Background

Atick *et al.* [1] proposed a method to use eigenheads to solve a *shape from shading* problem by leveraging the knowledge of object class, which was used to recover the shape of a 3D human face from single photograph. This line of research (including that of many others) ultimately culminated in the seminal work of Blanz & Vetter [2], who formulated an optimization problem to reconstruct a textured 3D face from one or more photographs in the context of inverse rendering. Though originally targeted to the computer graphics community, it did not take long for face recognition researchers to take note of the many advantages of this type of approach, especially its 3D representation framework – *ie.* shape surface-mesh + 2D texture-map. Our formulation is similar in essence. However, our implementation of various stages is more robust and amenable to efficient realization (*eg.* in hardware), depending on the particular application scenario.

Specifically, Vetter & Blanz [21] used a 3D variant of a gradient-based optical flow algorithm to derive the necessary point-to-point correspondence. Their method also employs color and/or texture information acquired during the scanning process. This approach will not work well for faces of different races or in different illumination given the inherent problems of using static textures. We present a simpler method of determining correspondences that does not depend on the color or texture information.

3. Face Model

Any parameterized 3D face model can be used together with the proposed shape recovery algorithm described in Section 4. However, the model parameters should describe well the silhouette contours of a real person's face for accurate matching. Thus, rough 3D mesh models in a relatively low resolution will not be adequate for this purpose. Our underlying face model is not synthetic but is based on real human faces measured by laser-based cylindrical scanners. This data-driven face model is limited in its expressive power by the number and variety of the faces in the training database. However, it can be easily expanded by incorporating new faces into the existing database.

3.1. Face Database Preprocessing

Our face database comes from USF dataset [18] and consists of Cyberware scans of 97 male adult and 41 female adult faces with various races and ages. The number of points in each face varies approximately from 50,000 to 100,000. All faces in the database were resampled to obtain point-to-point correspondence and then aligned to a reference face to remove any contamination of the PCA caused by pose variation and/or misalignment. The method we used to obtain the point-to-point correspondence is composed of the following steps:

1. Select a reference face F_r, which is the closest face to the mean face in the database and choose m feature points on it manually (in our case, we used $m = 26$). Let the position of the feature points be $\mathbf{q_{r,k}} \quad (k = 1..m)$.
2. For a face F_i in the database, select m feature points corresponding to the feature points on F_r. Let the position of the feature points be $\mathbf{q_{i,k}}$.
3. Deform F_r so that it fits the target face F_i. This requires the interpolation of all points in F_r under the constraint $\mathbf{q_{r,k}} = \mathbf{q_{i,k}}$. Let the deformed face be F_i^d. Now F_i^d has a shape similar to F_i since both have same locations for the all m feature points. Note that F_i^d has exactly the same number of points as F_r.
4. For each point in F_i^d, sample a point on the surface of F_i in the direction of underlying cylindrical projection (as defined by the scanner configuration). Now each resampled point on F_i has a corresponding point on F_r.
5. Repeat step 2 through step 4 for all F_i's $(i \neq r)$ in database to get the correspondences among all the faces in the database.

For step 3, a standard model for scattered data interpolation can be exploited [10, 12]. Note that, at step 4, we cannot get corresponding samples on the surface of F_i for some points on the boundary of F_i^d. It is likely that the two faces under consideration do not match exactly on the boundary. We keep track of the indices of those void sample points

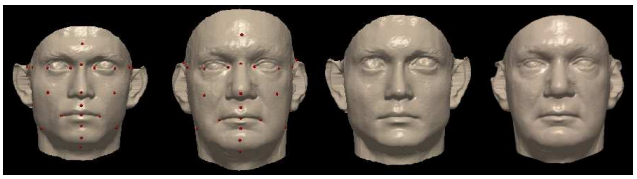

Figure 2: Getting correspondence between two faces. From left to right, reference face, target face, warped reference face, resampled target face. Note the void samples in the ears of the resampled target face.

and use only sample points whose indices are not void in any resampling of F_i in the database. Figure 2 depicts the process to establish the correspondence between reference and target faces.

We next applied PCA to the newly registered and aligned database of 3D faces faces to obtain our prior statistical shape model. As a consequence of this analysis, we can define all possible face geometries with *eigenheads* [1]. This decomposition can be used to reconstruct a new or existing face through the linear combination of eigenhead basis functions. Therefore, our face model is given by

$$H(\boldsymbol{\alpha}) = \boldsymbol{h_0} + \sum_{m=1}^{M} \alpha_m \boldsymbol{h_m} \tag{1}$$

and the model parameter is $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_M\}$, where $\boldsymbol{h_m}$ is the m^{th} eigenhead and $\boldsymbol{h_0}$ is the mean or average head.[1]

4. Shape-from-Silhouettes

In this section, we describe fitting our model parameters to a set of input silhouettes (taken from either a single video or a multi-camera rig setup). In Section 4.1 we describe the rationale behind our non-linear optimization technique, specifically for direct binary silhouette matching with a cost function, described in Section 4.2, which has no gradient information. This silhouette matching metric, by virtue of its simple yet effective design (boundary-weighting) is ideal for matching *partial* occluding contour segments and is relatively immune to noise/clutter and incomplete silhouettes (with "holes") obtained by background-subtraction. In Section 4.3, we describe how (easily) the texture can be extracted (lifted) from the source images (and blended or averaged) once the 3D shape is recovered. Note the contrast here to other work in this area where "eigenfaces" of the facial texture-maps are also used as model priors. We believe that given sufficiently high-quality inputs (and/or many frames), there is often no real need to estimate the texture-map.

[1]Note that while PCA is typically applied to the 2D facial texture as well (as in [2] for example) an "eigenface" model is neither required nor in fact necessarily desired in our framework (see Sections 4.3 and 7).

4.1. Nonlinear Optimization

Let $M(\boldsymbol{\alpha})$ be any arbitrary face model which produces a polygon mesh given a vector parameter $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \cdots, \alpha_n\}$. Let $S^k_{input}, k = 1..K$ be k^{th} input silhouette image. Also, let T be a similarity transformation that aligns a reference model face to the real 3D face. Then, $S^k_{model}(\boldsymbol{\alpha})$ is a silhouette image rendered by projecting $T(M(\boldsymbol{\alpha}))$ onto an image plane using the pose information appeared in the k^{th} silhouette image.

Provided we define a cost function f that measures the difference of two silhouette images, our goal is to find $\boldsymbol{\alpha}$ that minimizes the total penalty

$$E(\boldsymbol{\alpha}) = \sum_{k=1}^{K} f(S^k_{input}, S^k_{model}(\boldsymbol{\alpha})) \tag{2}$$

for a suitable cost function f.

We use *downhill simplex method* to minimize Eq.(2). This optimization process depends on the characteristics of the model parameter. Here, we discuss the optimization process based on our model parameter described on Section 3. Among the 137 eigenheads, we chose the first 60 eigenheads to reconstruct a 3D face. Furthermore, we found this number to be sufficient to capture most of the salient features in a human face. Thus, the corresponding coefficients serve as our multi-dimensional optimization parameter of dimensionality 60.

The simplex method can be easily adapted to our multi-dimensional face model. The initial simplex of 60 dimensions consists of 61 vertices. Let the coefficients $\boldsymbol{\alpha} = \{0, \cdots, 0\}$ (corresponding to the mean face) be one of the initial points $\mathbf{p}_0$ of the simplex. We can choose the other remaining 60 points to be

$$\mathbf{p}_i = \mathbf{p}_0 + \mu_i \mathbf{e}_i, \quad i = 1..60,$$

where $\mathbf{e}_i$'s are 60 unit vectors and μ_i can be defined by the characteristic length scale of each component of $\boldsymbol{\alpha}$. We set $\mu_i = 3\sqrt{\lambda_i}$, where λ_i is the i^{th} eigenvalue corresponding to i^{th} eigenhead in our face model. With this initial configuration, the movement of this 60 dimensional simplex is confined to be within our face space and there is no need to perform exhaustive searches in the exterior of the face space.

4.2. Silhouette Matching Metric

Now, we discuss how we design the cost function f in Eq.(2). The easiest way to measure difference of two binary images is the number of 'on' pixels when pixel-wise XOR operation is applied to the two images [7]. In this case,

$$f(S^k_{input}, S^k_{model}(\boldsymbol{\alpha})) = \sum_{i}^{H} \sum_{j}^{W} c(i,j) \tag{3}$$

$$c(i,j) = \begin{cases} 0 & \text{if } S^k_{input}(i,j) = S^k_{model}(\boldsymbol{\alpha})(i,j) \\ 1 & \text{otherwise.} \end{cases}$$

If our goal requires that $f = 0$, that is, if two silhouettes overlap exactly, the optimal solution will be unique in terms of $S^k_{model}(\boldsymbol{\alpha})$. However, if our objective function f cannot be reduced to zero given inherent characteristics of the problem, it is likely that there are multiple optimal solutions. Any preference among those multiple optimal solutions should be incorporated in the cost function.

In our case, the input silhouette area covers the full head including hair and the back, while our face model includes the front of the face delineated by the ears on the sides and lower part of the forehead from the top. Thus, our objective function, f, is often non-zero (or $f > 0$) since the silhouette generated by our model ($S^k_{model}(\boldsymbol{\alpha})$) considers only a partial area of the input silhouette (S^k_{input}) (see Figure 5). If we use the objective function f in Eq.(3), we could have multiple set of $S^k_{model}(\boldsymbol{\alpha})$ that minimize f and we cannot guarantee that these solutions match the real boundary contours in the input silhouettes. Our goal is to match the real boundary contours between input and model silhouettes and f is required to be the global minimum. Accordingly, we impose higher penalty for the mismatch near the boundary pixels of input silhouettes.

Though a mismatch in the pseudo contour area contributes a higher cost to f, this contribution can be considered as a constant factor. Our new cost function replaces $c(i,j)$ in Eq.(3) with

$$c(i,j) = \begin{cases} 0 & \text{if } S^k(i,j) = S^k_m(\boldsymbol{\alpha})(i,j) \\ \frac{1}{d(i,j)^2} & \text{otherwise} \end{cases} \quad (4)$$

$$d(i,j) = D(S^k)(i,j) + D(\tilde{S}^k)(i,j),$$

where $D(S)$ is the Euclidean distance transform of binary image S and $\tilde{S}$ is the inverse image of S. Note that d represents a distance map from silhouette contour and can be computed once in a preprocessing step. We call this cost function *boundary-weighted XOR* , which provides a simple and effective alternative to precise contour matching schemes. As a result, there is no need for expensive operations of correspondence, edge-linking, curve-fitting, distance computations between boundary curves; all needed when precise contour matching schemes are used. Thus, our optimization algorithms are fast and robust.

4.3. Texture Lifting

Our optimized 3D model matches all input silhouette images as close as possible. Since the input silhouette images are obtained from the corresponding texture images, we do not need any further registration process for texture extraction. We extract texture colors in object space rather than image space and do not create a single texture-map image. That is, for each 3D vertex in the reconstructed 3D face, we assign a color value which is determined from multiple texture images. To do so, we proceed as follows.

Our approach is a view-independent texture extraction approach [12, 15, 19]. Each vertex is projected to all image planes and tested if the projected location is within the silhouette area and if the vertex is visible (not occluded) at each projection. For all valid projections, we compute the dot product between the vertex normal and the viewing direction, and use the dot product as a weight of the texture color sampled at the projected image location. The final color value at a vertex is computed by dividing the weighted sum of texture values of all valid projections by the sum of weights.

5. Sensing Geometry

As already noted, we have experimented with two nearly equivalent operational scenarios in the face acquisition system's sensor geometry. Firstly, a multi-camera rig which takes static but multiple "snapshots" of the subject in a relatively controlled and instrumented sensing environment, best suited for initial model-building, training and algorithm development. Secondly, lower-cost and less-constrained single stream monocular video wherein either the subject is static and the camera is moving or where the camera is stationary and the subject is moving. This latter scenario is more in line with real-life applications where often equipment costs and the need for covert operation exclude highly "visible" multi-camera rigs. Our own view is to use the multi-camera rig in the laboratory for testing and model building (*ie.* eigenhead PCAs) and then to apply these in operational settings with single/fewer cameras.

5.1 Multi-Camera Rig

We now describe how the proposed silhouette matching scheme is applied to the silhouette images taken by multiple static cameras simultaneously from different viewpoints (here we are assuming that all cameras are calibrated). As with the visual hull, it is important to choose the viewpoints carefully to get maximally informative 3D shape information from a set of silhouette images. After some preliminary experimentation with various geometries one candidate arrangement was selected for the majority of the experiments. We should note that the optimal arrangement of N cameras is currently an ongoing part of our research project. In particular, the eleven camera positions that we used were sampled on the front hemisphere around the face/head as shown in Figure 3. These were found to be adequate given our physical constraints with the rig as well as shape capture accuracy. A more detailed exploration of different camera placements and the subsequent impact on our system's overall reconstruction accuracy is published elsewhere.

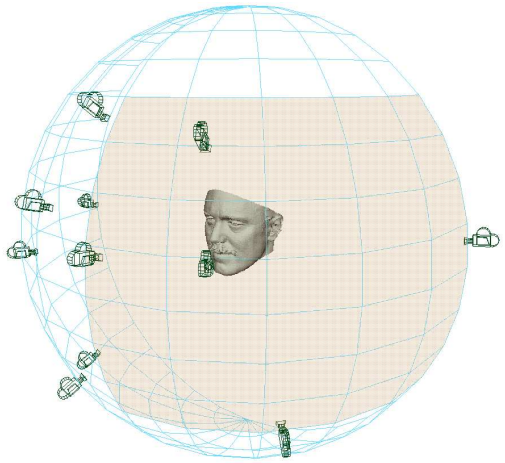

Figure 3: The layout of 11 cameras corresponding to the actual multi-camera rig (used in both synthetic and real experiments).

Though we assume all cameras are calibrated, we still need to align our model to the estimated pose (rigid transformation, T) of a real person's head defined in the same coordinate system with the multiple cameras. Finding the alignment transformation T is not trivial using only the silhouette information. The form of T depends on the pose and size of the face of a person to be captured. T can be defined as

$$T(\mathbf{x}) = s(\mathbf{R}\mathbf{x} + \mathbf{t}),$$

where s is a scale factor, $\mathbf{R}$ is a rotation matrix, $\mathbf{t}$ is a translation vector. The alignment problem is then one of minimization of the functional:

$$\sum_{j=1}^{L} \|\mathbf{x}_j - T(\mathbf{y}_j)\|^2, \tag{5}$$

in terms of s, $\mathbf{R}$ and $\mathbf{t}$. It should be noted that $\mathbf{x}_j$ is the j^{th} 3D feature point in real face, $\mathbf{y}_j$ is the j^{th} 3D feature point in a reference model face and L is the number of feature points to be used.

We already know $\mathbf{y}_j$. However, $\mathbf{x}_j$ is determined from a standard non-linear least square minimization technique [13, 17]. A Levenberg-Marquardt algorithm is applied to obtain the 3D point locations that correspond to L feature points selected manually in a small number of (3-4) texture images. We used $L = 7$ in our experiments. Once we determine $\mathbf{x}_j$, then, we compute the values of s, $\mathbf{R}$ and $\mathbf{t}$ such that Eq.(5) is minimized. The needed parameters are obtained from an application of the *full ordinary Procrustes analysis* [4].

5.2 Monocular Sequence

We consider a video sequence captured in front of a fixed video camera (*eg.* a $320 - \times - 240$ webcam). The user is required to start from a frontal view and then allowed to rotate his or her head arbitrarily to capture face shapes from various viewing angles. For the first frame, the user is required to select two outer eye corners and two mouth corners to locate a 3D mask (template) to the face area.[2] The mask is used to track the facial area in the subsequent frames by its rigid transform. After finding six-dimensional motion parameters for each frame, we extract $M(\ll N)$ mostly significant poses and generate corresponding silhouette images by background subtraction. For those M silhouette images, the shape parameters are adjusted so that the face model with the fitted shape parameters provides the closest silhouette shapes to the given input silhouettes.

In this section, we describe how we can estimate poses for a video sequence with arbitrary head rotation. We start from the well-known optic flow constraint. Let $\mathbf{u} = (u, v)$ be a pixel location in a frame and $\mathbf{u}' = (u', v')$ be the corresponding pixel location in the next frame. Then, we assume the pixel intensities at $\mathbf{u}$ and $\mathbf{u}'$ do not change, that is, $I(\mathbf{u}, t) = I(\mathbf{u}', t + 1)$. If a specific region in a image (e.g. human head) is governed by a rigid transform, we can define a mapping function from a 3D point $\mathbf{p} = (x, y, z)$ to a 2D pixel location $\mathbf{u}$ such that $\mathbf{u} = P(f(\phi, \mathbf{p})) = P(\mathbf{R}\mathbf{p} + \mathbf{t})$, where P represents a projection model, $\phi = (R_x, R_y, R_z, t_x, t_y, t_z)$ is a 6-dimensional motion parameter, $\mathbf{R}$ is a rotation matrix and $\mathbf{t}$ is a translation vector. Let ϕ is a motion parameter of a head at time t and ϕ' be a motion parameter to be recovered at time $t + 1$. Then, our goal is to find ϕ' that minimizes

$$g(\phi, \phi') = \sum_{\mathbf{p} \in \mathrm{G}} \|I(P(f(\phi, \mathbf{p})) - I(P(f(\phi', \mathbf{p}))\|^2 \tag{6}$$

where G is a set of visible points on the 3D face template in both frames. We use the mean face of a 3D face database as our template face and call it a *mask* in the following sentences. Since the first-order Taylor expansion of g is non-linear with respect to ϕ', it is not trivial to apply the well-known Lucas-Kanade method [8] based on Newton-Raphson style iteration using spatial and temporal image gradients. We exploit a perspective projection, which is more proper model to deal with the sequences taken at relatively short distance between head and camera. Instead of using gradient information, we apply *downhill simplex* method, to minimize Eq. (6) using cost function evaluations only [13].

The optimization tracking process can be easily misled due to noise if we only use information from the previous

[2] Based on extensive past experience, we are confident that automatic face/eye-detectors can be put to good use here to initialize the mask (clearly the preferred mode of operation) or at least correct for any drift in tracking.

frame to estimate the motion of the current frame. Instead of using the single reference frame, we use a weighted combination of Eq. (6) with multiple reference frames that are already registered. A proposed objective function is

$$c(\phi') = w_1 g(\phi, \phi') + w_2 g(\phi_n, \phi') + w_3 g(\phi_i, \phi') \quad (7)$$

where ϕ_i is the initial pose and ϕ_n is the pose that is already registered and the closest to ϕ in terms of L_2 norm. The weights w_2, and w_3 are determined by the pose distance $\|\phi - \phi_n\|$ and $\|\phi - \phi_i\|$ respectively and $w_1 = 1 - w_2 - w_3$. In this way, we also prevent the estimation error from being accumulated through the frame-by-frame basis optimization approach.

6 Experiments

We first touch on some implementation issues which also high-light some of the advantages of our particular scheme. One concern is the speed of the optimization process. The most time-consuming part in a function evaluation is the silhouette generation part. Since our face model is of very high resolution (approximately 48000 vertices and 96000 triangles), even rendering with flat shading takes considerable time when it should be repeated in an optimization process. A simple remedy for this problem is to reduce the mesh resolution by vertex decimation. Also, if we reduce the mesh resolution, it is natural to reduce the resolution of silhouette images accordingly (originally 1024×768). The reduction in model and image resolution will accelerate the XOR computation process. In our experiments, we determined that 95% decimation in the mesh and 50% reduction in image resolution resulted in a similar convergence rate and a lower (1/10) cost required for original resolution data.

Another way to expedite the optimization process is to employ a hierarchical approach [7]. For example, 99% decimation in mesh resolution and a 75% reduction in image resolution resulted in only 30-40 seconds until convergence. As a result, it is likely better results can be obtained than those obtained using only high resolution data. All the results presented here were obtained from this hierarchical optimization technique. Note that the shape parameters ($\boldsymbol{\alpha}$) are not directly dependent on the input silhouette image resolution and do not dictate the 3D output mesh resolution. The manner of resolution-independence built into our scheme is a very desirable feature. Our statistical shape model already includes fine details which allows us to use lower-resolution sensing in the input images and XOR computations for faster shape recovery.

Fortunately a wealth of synthetic data can be generated from our face model directly by sampling the implicit Gaussian distribution inherent in the PCA "eigenhead" model. For example, to illustrate the robustness of our shape reconstruction method, we chose 50 individual faces from the database and 50 synthetic faces generated by randomly sampled parameters, $\alpha_i = (-0.8\sqrt{\lambda_i}, 0.8\sqrt{\lambda_i}), i = 1...60$, according to the prior Gaussian distribution. Eleven synthetic cameras were then positioned in the front hemisphere around the subject (as in Figure 3). The input silhouette images were acquired by rendering each of the sample faces in the image planes of the 11 virtual cameras. Figure 4 shows resulting reconstructions from our optimization and shape refinement process. The selected faces in the figure correspond to the minimum, average, and the maximum L_2 error among all the 100 samples. We observe that our silhouette matching algorithm captures the most important features of a face within our constructed face space.

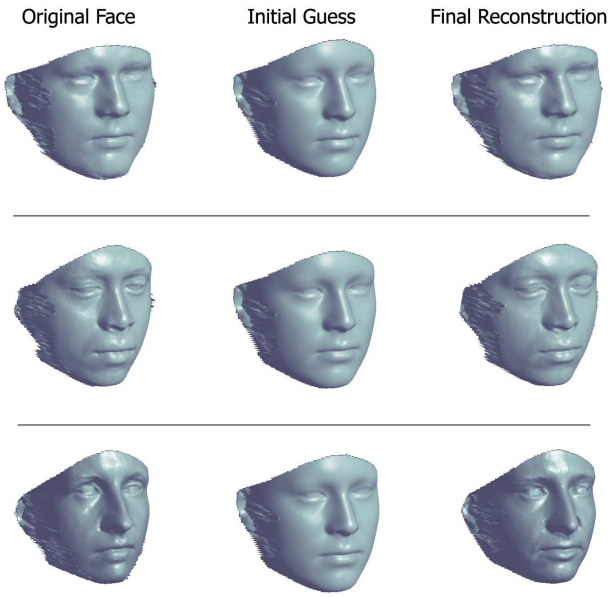

Figure 4: Reconstruction of synthetic faces: (top) minimum L_2 error, (middle) average L_2 error, (bottom) maximum L_2 error.

The challenges in using images taken by real cameras are in silhouette acquisition, accuracy of camera parameters, misalignment, and 'clutter' (excess head area beyond the face model). We assume that silhouette images can be easily acquired by a simple background subtraction technique. We calibrated the eleven static cameras (Figure 3) by a standard technique using a calibration object [17]. One could enhance this initial camera calibration by a technique that uses silhouette images [6, 14]. Figure 5 shows how our model face fits to real silhouette images of a Caucasian face. Note the similarity of alignment to the synthetic cases in Figure 4, demonstrating that our boundary-weighted XOR cost function allows the model-generated silhouette contour to closely fit the boundary of input silhouette images. Note that this alignment cannot be achieved with a simple XOR cost function due to the lack of preference for contours.

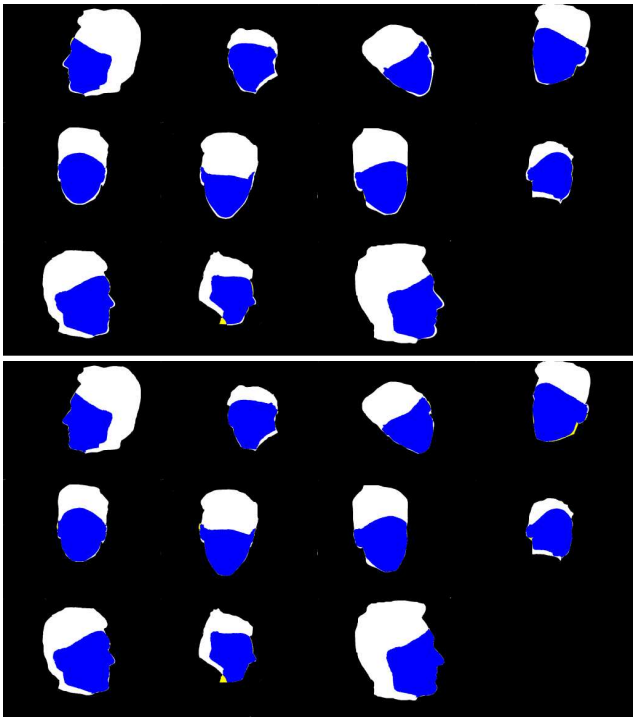

Figure 5: Real silhouette images of a Caucasian head before (above) and after (below) optimization with our face/head model. NOTE: the input silhouette is shown in white, its overlap with the model is shown in blue and areas where the model has no overlap with the input are shown in yellow.

Figure 6 demonstrates the effectiveness of 3D reconstruction and subsequent texture-mapping of the Caucasian in Figure 5. In addition, Figure 7 shows the results of the final 3D model captured from an Asian subject using the same technique. Note that the location of eyes and the shape of noses and lips in the texture-mapped images agree well with the reconstructed 3D geometry. It is remarkable that the race information, which is expected to be coupled with silhouette contour, was successfully captured by our silhouette matching scheme.

Initial experiments with 3D face tracking in monocular sequences using our formulation have yielded very promising results (especially in light of the many differences in our approach). Figure 8 shows some frames from a sequence with a stationary camera (moving subject) wherein an average-face 3D "mask" was used to track the 3D movement of the head. In the figure, this canonical mask is superimposed onto the selected frames, with holes "drilled" in at the eyes/mouth to help with visualizing the alignment. This sequence required only hand initialization of the eyes/mouth in the first frame. All subsequent mask/pose tracking was performed using the simplex optimization described in Section 5.2.

From this sequence, we automatically selected 11 frames

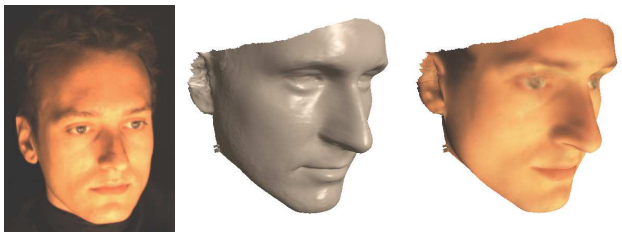

Figure 6: A rendering of the captured 3D model of the Caucasian in Figure 5. Left panel is a (real) image taken from our rig. Note: subject was *not* laser-scanned or included in the training set.

Figure 7: A rendering of the captured 3D model of an Asian subject. Left panel is a (real) image taken from our rig. Note: subject was *not* laser-scanned or included in the training set.

(deemed sufficiently distinct by means of silhouette "clustering" using their foreground shape moments) and used these frames as the equivalent virtual multi-camera views for reconstruction. A subset of these "calibrated" views is shown in Figure 9, where the background-subtracted input or foreground "silhouettes" (in white) are shown along with the back-projected model silhouettes. The model mask's intersection with the input is shown in blue, whereas yellow indicates the model silhouette regions with no overlaps (missing inputs). Note that despite the noise and partial silhouette data, the boundary-weighted XOR cost function nicely fits the model to the salient portions of the data (mostly contour) while ignoring spurious detail and clutter. This feature makes our silhouette cost function particularly desirable in real applications.

7. Discussion

We presented a robust and efficient method to capture (or reconstruct) 3D human faces from silhouettes. Few user-specified parameters are required making our method close to an automatic method. We proposed a novel algorithm for establishing correspondence between two faces using an efficient boundary-weighted XOR cost function for the optimization. This method is robust with partial heads or binary silhouettes with noise and clutter. Moreover, our method is relatively resolution-independent allowing for expedient reconstructions tailored for a given sensor and the use of coarse-to-fine optimization for computational speedup.

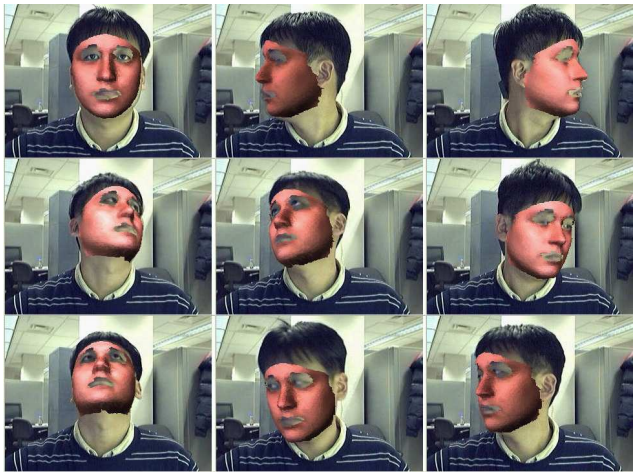

Figure 8: Pose estimation and tracking for a monocular sequence. The default (average) 3D "mask" is overlaid on the input video.

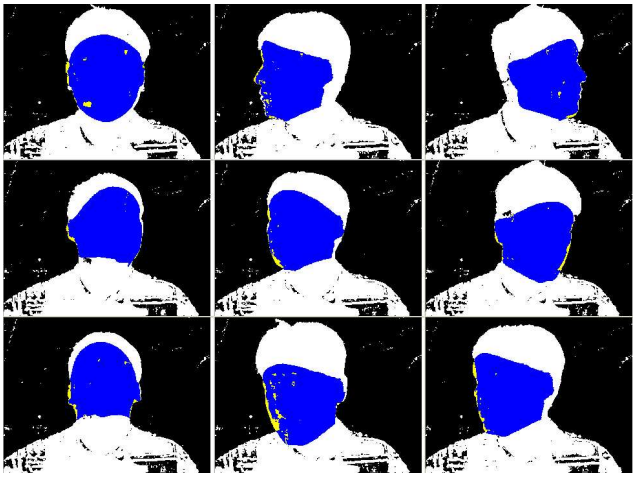

Figure 9: Shape fitting for a monocular sequence (white is input, yellow is the model, and blue is model-input intersection).

We believe that 3D face capture will figure prominently in the future of face recognition and to that end systems capable of decoupling the shape/texture from pose/illumination artifacts have a great advantage. For example, armed with the canonical texture-map alone, one can exploit the variety of 2D face-recognition systems already in existence (simply feed the "unwrapped" texture-map as input). Capturing the 3D shape model automatically obviates the problem of pose as typically encountered with 2D view-based methods. The areas of future research we are currently focusing on are the determination of the optimal number and placement of cameras (in the multi-camera setup), robust pose tracking in monocular sequences, improved texture "lifting" and blending for higher-quality textures, estimation and decoupling of the scene illumination from the texture-map and hence the removal of any such adverse impact on subsequent face recognition performance.

References

[1] J. J. Atick, P. A. Griffin, and N. Redlich, "Statistical Approach to Shape from Shading: Reconstruction of 3D Face Surfaces from Single 2D Images," *Neural Computation*, Vol. 8, No. 6, pp. 1321-1340, 1996.

[2] V. Blanz and T. Vetter, "A Morphable Model for the Synthesis of 3D Faces," In *Proceedings of SIGGRAPH 99*, July 1999.

[3] Cyberware, Inc., Monterey, CA. URL: http://www.cyberware.com/

[4] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis*. John Wiley & Sons, New York, 1998.

[5] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, The MIT Press, Cambridge, MA, 1993.

[6] A. A. Grattarola. Volumetric Reconstruction from Object Silhouettes: A Regularization Procedure. *Signal Processing*, Vol. 27, No. 1, pages 27-35, 1992.

[7] H. P. A. Lensch, W. Heidrich, and H. Seidel, "Automated Texture Registration and Stitching for Real World Models," In *Proceedings of Pacific Graphics '00*, October 2000.

[8] B. D. Lucas and T. Kanade. "An Iterative Image Registration Technique with an Application to Stereo Vision," In *Proc. of Imaging Understanding Workshop*, pp. 121-130, 1981.

[9] W. Matusik, C. Buehler, R. Raskar, L. McMillan, and S. J. Gortler. Image-Based Visual Hulls. In *Proceedings of SIGGRAPH 2000*.

[10] G. M. Nielson, "Scattered Data Modeling," *IEEE Computer Graphics and Applications*, Vol. 13, No. 1, pp. 60-70, January 1993.

[11] A. Pentland, B. Moghaddam, and T. Starner. View-Based and Modular Eigenspaces for Face Recognition. *Proc. of CVPR'94*, 1994.

[12] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin, "Synthesizing Realistic Facial Expressions from Photographs," In *Proceedings of SIGGRAPH 98*, July 1998.

[13] W. H. Press, B. P. Flannery, S. A. Teukolosky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, 1988. In Proceedings of The Third International Conference on Visual Computing

[14] P. Ramanathan, E. Steinbach, and B. Girod. Silhouette-based Multiple-View Camera Calibration. In *Proceedings of Vision, Modeling and Visualization 2000*, pages 3-10, November 2000.

[15] C. Rocchini, P. Cignoni, C. Montani, and R. Scopigno, "Multiple Textures Stitching and Blending on 3D Objects," In *Rendering Techniques '99 (Proc. 10th EG Workshop on Rendering)*, pp. 119-130, 1999.

[16] R. Szeliski. Rapid Octree Construction from Image Sequences. *CVGIP: Image Understanding*, Vol. 58, No. 1, pages 23-32, 1993.

[17] R. Szeliski and S. Kang. Recovering 3D Shape and Motion from Image Streams Using Non-Linear Least Squares. Technical Report, Robotics Institute, Carnegie Mellon University, March, 1993.

[18] USF DARPA HumanID 3D Face Database, Courtesy of Prof. Sudeep Sarkar, University of South Florida, Tampa, FL.

[19] M. Tarini, H. Yamauchi, J. Haber, and H.-P. Seidel, "Texturing Faces," In *Proceedings Graphics Interface 2002*, pp. 89-98, May 2002.

[20] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams under Orthography: a Factorization Method," *International Journal of Computer Vision*, Vol. 9, No. 2 pp. 137-154, Nov. 1992.

[21] T. Vetter and V. Blanz. Estimating Coloured 3D Face Models from Single Images: An Example Based Approach. In *Computer Vision - ECCV '98*, Vol II, Freiburg, Germany, 1998.