Model-based 3D Shape Recovery from Single Images of Unknown Pose and Illumination using a Small Number of Feature Points

Ham M. Rara and Aly A. Farag  
CVIP Laboratory, University of Louisville  
{hmrara01, aafara01}@louisville.edu

Todd Davis  
EWA Government Systems, Inc.  
tdavis@ewa.com

Abstract

This paper proposes a model-based approach for 3D facial shape recovery using a small set of feature points from an input image of unknown pose and illumination. Previous model-based approaches usually require both texture (shading) and shape information from the input image in order to perform 3D facial shape recovery. However, the methods discussed here need only the 2D feature points from a single input image to reconstruct the 3D shape. Experimental results show acceptable reconstructed shapes when compared to the ground truth and previous approaches. This work has potential value in applications such as face recognition at-a-distance (FRAD), where the classical shape-from-X (e.g., stereo, motion and shading) algorithms are not feasible due to input image quality.

1. Introduction

Humans have the uncanny ability to perceive the world in three dimensions (3D), otherwise known as depth perception. The amazing thing about this ability to determine distances is that it depends only on a simple two-dimensional (2D) image in the retina [10]. Psychologists approach this problem by asking what information is available from the 2D image that enables humans to perceive depth. This is referred to as the cue approach to depth perception. According to the cue theory [3], humans learn the connection between this cue (e.g., binocular cues, motion-related cues, lighting and shading cues) and depth through previous experiences.

In computer vision, there is an analogous term to depth perception, which is shape recovery. It is the computational equivalent of the depth perception concepts from psychology [24]. The class of algorithms that deals with different types of shape recovery is conveniently named as shape-from-X techniques, where X can be stereo, motion, shading, texture, etc.

The shape-from-shading problem (SFS) is an interesting subject in computer vision that involves recovering the 3D shape of an object using the cues of lighting and shading. Previous works have shown that constraining the shape-from-shading algorithm to a specific class of objects can improve the accuracy of the recovered 3D shape [15]. There is particularly a huge interest in the 3D shape recovery of human faces from intensity images. Zhao and Chellapa [25] used the known bilateral symmetry of frontal faces as a geometric constraint in their approach.

Atick et al. [4] proposed the first statistical SFS method by parameterizing the set of all possible facial surfaces using principal component analysis (PCA). Smith and Hancock [22] embedded a statistical model of surface normals within a shape-from-shading framework.

There has been a substantial amount of work regarding statistical face processing in the computer vision literature. The morphable model framework of [5] estimates the shape and texture coefficients from an input 2D image, together with other scene parameters, using an optimization method based on stochastic gradient descent. It is a 3D extension of the seminal work of Cootes et al. [7] on Active Appearance Models (AAM), where a coupled statistical model is generated to describe the 2D shape and appearance (albedo).
information of faces.

Castelan et al. [6] developed a coupled statistical model, which is a variant of the combined AAM [7], that can recover 3D shape from intensity images with a frontal pose. The shape and intensity models in Castelan’s work is similar to that of the AAM model. Note that in the shape recovery literature, albedo can be used, interchangeably, with the term intensity. The primary difference in Castelan’s approach is that the 2D shape model in AAM is replaced with a 3D shape (height map) model [9].

Recently, Rara et al. [19] formulated a 3D shape recovery method, by modifying the framework of [6] to handle images of general lighting. This approach, which uses only few matrix operations for shape reconstruction, provides a computationally efficient alternative to the iterative methods of [18][12], while offering comparable results.

The methods mentioned above require both texture (shading) and 2D shape information from the input image to perform 3D reconstruction. The goal of this work is to come up with a model-based approach that can reconstruct 3D shape using only the 2D feature points from an input image but with comparable results to its predecessors, which use both shading and 2D shape information. This work has potential value in applications such as face recognition at a distance (FRAD) [17], where the classical shape-from-X (e.g., stereo [16], motion [13] and shading) algorithms are not feasible due to input image quality.

The rest of the paper is organized as follows: Section 2 discusses basic definitions and notations. Section 3 talks about the two related methods to the proposed algorithm. Section 4 describes the proposed approach itself. Section 5 concludes the paper and mentions the anticipated future work.

2. Basic Definitions and Notations

The geometry of a face is represented as a shape vector that contains the XYZ coordinates of its vertices, i.e., $S = (X_1, Y_1, Z_1, \ldots, X_n, Y_n, Z_n)^T$. The shape model can be constructed using a data set of $m$ samples.

It is computationally convenient to reduce the dimensionality of the shape space, especially when dealing with high-dimensional shape vectors. Using Principal Component Analysis (PCA) on the data matrix provides us with $m - 1$ eigenvectors $S_i$, their corresponding eigenvalues (variances) $\sigma_i^2$, and the mean shape $\bar{s}$. New shapes $s$ can be derived from an equivalent model, i.e.,

$$s = \bar{s} + \sum_{i=1}^{m-1} a_i S_i$$  \hspace{1cm} (1)

where $a = (a_1, \ldots, a_{m-1})^T$ is the shape parameter vector. In matrix notation, the above equation can be expressed as $s = \bar{s} + Sa$.

A realistic 2D face can be generated from the 3D shape produced by the PCA model [20], i.e.,

$$s_{2D} = fPR(s + t_{3D}) + t_{2D}$$  \hspace{1cm} (2)

where $t_{3D}$ and $t_{2D}$ are concatenated translation vectors of length $3N$ and $2N$, respectively. $R$ is a $3N \times 3N$ block diagonal matrix which performs the combined rotation $R_xR_yR_z$ for all $N$ vertices. $P$ is a $2N \times 3N$ orthographic projection matrix for the set of vertices. Note that only a subset of the original set of vertices will be visible. The z-buffer algorithm [8] is used for hidden-surface removal.

The same concepts above are expressed in a different terminology in [11]. 2D shape is represented as $x_i \in \mathbb{R}^4$, and the corresponding 3D shape as $X_i \in \mathbb{R}^3$, using homogeneous coordinates. The goal is to find the $(3 \times 4)$ camera projection matrix $C$, i.e., $x_i = CX_i$. Assuming an affine camera, the camera projection matrix $C$ can be solved using the Gold Standard Algorithm (Algorithm 7.2) in [11].

3. Related Methods

Previous methods that perform face reconstruction from a small number of feature points can be classified into two groups, namely: (a) iterative and (b) linear approaches (non-iterative).

3.1. Iterative Methods

The algorithms in [23] and [21] use the 3D-to-2D projection equation of (2). They replace the 3D shape $s$ with the shape model from PCA $(s = \bar{s} + Sa)$ to allow minimization with respect to the shape parameter $a$, i.e.,

$$s_{2D} = fPR(s + Sa + t_{3D}) + t_{2D}$$  \hspace{1cm} (3)

where $f$ is a scale parameter, $P$ is an orthographic projection matrix, $R$ is the rotation matrix, $t_{2D}$ and $t_{3D}$ are translation vectors in 2D and 3D, respectively.

Notice that if the rendering process is inverted, the shape parameters $a$ can be recovered from the shape error. As long as $f$, $P$, and $R$ are kept constant, the relation between the shape $s_{2D}$ and $a$ is linear, i.e.,

$$\frac{\partial s_{2D}}{\partial a} = fPRS$$  \hspace{1cm} (4)

The above equation comes from differentiating $s_{2D}$ with respect to $a$ in (3). Transferring $\partial a$ to the other side of the equation, we have a linear system, $\partial s_{2D} = fPRS(\partial a)$. Therefore, given the shape error $\partial s_{2D}$, estimated by the displacement of a set of feature points, the update of $a$ can be determined.
3.2. Linear Methods

Recently, Aldrian et al. [2] proposed a shape recovery method that can extract 3D facial shape using only a sequence of matrix operations. The 2D projection of the 3D feature points in the shape model is referred to as \( y_{mod2D,i} \). The 2D feature points in the input image are denoted as \( y_i \).

Instead of minimizing with respect to the shape parameter \( a \) in (1), the method in [2] minimizes with respect to a related variable, namely, the variance normalized shape parameter \( c_s = (\frac{a_1}{\sigma_1}, \cdots, \frac{a_m}{\sigma_m}) \), where \( \sigma_i \) are the eigenvalues after PCA is performed on the shape data matrix.

The next step is to discuss the error functional [2] to be minimized in only a single step. This can be done by differentiating the functional, setting it to 0, and solving for the error functional is

\[
E = \sum_{i=1}^{3N} \frac{[y_{mod2D,i} - y_i]^2}{\sigma_{2D,i}^2} + ||c_s||^2
\]

where \( \sigma_{2D,i}^2 \) is the 2D point error variance that explains the difference between the observed and modeled feature point positions in the input image. The value of \( \sigma_{2D,i}^2 \) is determined after performing some offline training. The closed-form solution for \( c_s \) is

\[
c_s = [diag(w_i + 2)V^T]^{-1}V^T T_2
\]

The steps on how to setup the constants \( w_i \) and matrices \( V \) and \( T_2 \) are found in [2]. Therefore, using only a sequence of matrix operations, the normalized shape parameters \( (c_s) \) can be computed given the location of the 2D feature points from the input image, as well as the camera projection matrix \( C \). It is straightforward to compute the actual shape parameter \( a \), i.e., \( a_i = c_s, i \sigma_i \).

4. Proposed Approach

This section proposes a non-iterative, model-based approach for face reconstruction using a small set of feature points from an input image of unknown pose and illumination. The proposed algorithm formulates the shape recovery problem into a regression framework, i.e., it uses Principal Component Regression (PCR) to reconstruct 3D shape.

Fig. 1 illustrates this problem succinctly. The input is a 2D image with annotated feature points. Only the information from the feature points is given to a 3D estimation black box. The output can be one of the two cases: (a) 3D sparse shape and (b) 3D dense shape.

The USF database [1] used in this work contains both albedo and dense shape, where they are expressed as Monge patches, i.e., \((x, y, Z(x, y), T(x, y))\). The image data of the USF database samples are annotated with 76 points, which are related to the important features of the face. Since both image and dense shape data are in correspondence with each other, the annotation points can also be applied to the height maps, which results into 3D sparse shapes. Since we have multiple USF subjects, we have a series of dense shapes together with corresponding sparse shapes, as illustrated in Fig. 2. This series of dense and sparse shapes is integral to the proposed method in this work.

4.1. Case I: Output 3D Sparse Shape

Suppose we have an input 2D sparse shape and the goal is to find the camera projection matrix \( C \) from its unknown (and yet to be solved) actual 3D sparse shape. A good substitute for this unknown 3D shape is the mean shape. A camera projection matrix can be computed between the mean 3D sparse shape and the input 2D sparse shape. Further, this projection matrix can be used to project a sample USF 3D sparse shape to the 2D space. The projection matrix \( C \) can be used to project all USF database samples to the 2D space, as illustrated in Fig. 3.

Notice that Fig. 3 is an example of a combined model.
coefficients and yields, \( x \). The least squares method then provides

\[
\text{OUTPUT}(\text{MLR}) \text{ is then performed between the low-dimensional representations of } x_i \text{ and } X_i. \]

Let \( T = [b_{s2D,1}, \ldots, b_{s2D,m-1}] \) and \( U = [b_{s3D,1}, \ldots, b_{s3D,m-1}] \) be the low-dimensional representations of \( x_i \) and \( X_i \), respectively. Performing MLR yields, \( U = TC_R + F \), where \( C_R \) is the matrix of regression coefficients and \( F \) is the matrix of random noise errors. The least squares method then provides

\[
\tilde{C}_R = (T^T T)^{-1} T^T U \tag{7}
\]

There are two remaining steps before the 3D sparse shape can be recovered. The shape coefficient of the 2D input feature points need to be solved, i.e., \( b_{s2D,inp} = P^T_{s2D}(x_{inp} - s_{2D}) \). Using the PCR model above, the 3D sparse shape coefficient can be inferred with the following equation, \( b_{s3D} = b_{s2D,inp} C_R \). The solved shape coefficient \( b_{s3D} \) can be substituted to the 3D shape model, i.e., \( x_r = \bar{s}_{3D} + P_{s3D} b_{s3D} \), to get the desired output. Algorithm 1 below summarizes these steps.

This section will compare the proposed approach above to recent methods that deal with the same problem, namely, the iterative approach of [23] and the linear (non-iterative) contribution of [2]. Fig. 4 shows the recovered 3D sparse shape of the same input using the three algorithms, together with the ground truth. Fig. 4 (View 2) represents the projection of the recovered 3D shapes to the \( x-y \) plane. Notice that the results of the proposed method and that of Aldrian et al. [2] are similar.

The next point of comparison will be the timing results of the three algorithms under the same computational conditions. Fig. 5(a) presents the side-by-side stem plot of the time (in seconds) needed to recover the 3D sparse shape for inputs generated from 80 samples of the USF database. Notice that the proposed method is computationally faster than the others, due to its simplistic regression framework. The method of [2] requires significant time in offline training for the 2D error variance \( \sigma_{2D,i}^2 \) in (6), which is not included in Fig. 5(a).

The ultimate goal in the previously discussed methods is to recover 3D shape given only 2D input points. The last point of comparison for this section is the 3D shape error. Let \( S_{err}^{3D} \) be the recovered 3D shape and \( S_{gt}^{3D} \) be the ground truth shape. The 3D shape error is simply the norm of the difference between the recovered and true shapes, i.e.,

\[
S_{2D}^{err} = \|S_{3D}^{err} - S_{3D}^{gt}\| \tag{8}
\]

Fig. 5(b) presents the side-by-side stem plot of the 3D shape error (8) for inputs generated from 80 samples of the USF database. Similar to the timing results, both the proposed method and that of Aldrian et al. [2] outperform than Zhang et al. [23].

There are five conclusions that can be drawn from these experiments, the proposed method: (a) is competitive due to its linear and non-iterative nature, (b) does not need explicit training, as opposed to that of [2], (c) has comparable results to [2], at a shorter computational time, (d) better in all aspects than Zhang and Samaras [23], and (e) has the limitation, together with [2] and [23], in terms of the need to manually annotate the input 2D feature points. This limitation can be easily alleviated by using automatic facial fea-

---

**Algorithm 1** Principal Component Regression (PCR) Framework for 3D Sparse Shape Recovery

**INPUT:** (a) Input image feature points, \( x_{inp} \) (b) USF sparse shape samples: \( \{x_1, \ldots, x_n\} \) (c) Sparse mean shape, \( X_m \)

**OUTPUT:** (a) Recovered 3D sparse shape, \( x_r \)

1. **Solve for the camera projection matrix:** Determine \( C \) such that \( x_{inp} = CX_m \).
2. **Project all 3D sparse shapes to the 2D space using the computed projection matrix:** Solve for \( x_i = CX_i \), such that \( x_i = CX_i \).
3. **Build the 3D sparse shape model from the USF samples using PCA:** Construct \( s_{3D} = \bar{s}_{3D} + P_{s3D} b_{s3D} \).
4. **Build the 2D sparse shape model from the projected 2D USF samples \( \{x_1, \ldots, x_n\} \):** Construct \( s_{2D} = \bar{s}_{2D} + P_{s2D} b_{s2D} \).
5. **Replace the 3D shape samples \( \{x_1, \ldots, x_n\} \) with its coefficients:** Solve for \( b_{s2D,inp} = P^T_{s2D}(x_{inp} - \bar{s}_{2D}) \).
6. **Replace the projected 2D shape samples \( \{x_1, \ldots, x_n\} \) with its coefficients:** Solve for \( b_{s2D,inp} = P^T_{s2D}(x_{inp} - \bar{s}_{2D}) \).
7. **Setup matrices for Principal Component Regression (PCR):** Let \( T = [b_{s2D,1}, \ldots, b_{s2D,m-1}] \), and \( U = [b_{s3D,1}, \ldots, b_{s3D,m-1}] \).
8. **Build the PCR model:** Construct \( \hat{C}_R = (T^T T)^{-1} T^T U \).
9. **Solve for the shape coefficients of the 2D input feature points \( x_{inp} \):** Solve for \( b_{s2D,inp} = P^T_{s2D}(x_{inp} - \bar{s}_{2D}) \).
10. **Solve for the shape coefficients:** Get \( b_{s3D} = b_{s2D,inp} C_R \).
11. **Solve for the recovered 3D sparse shape:** \( x_r = \bar{s}_{3D} + P_{s3D} b_{s3D} \).
Figure 4. Recovered 3D sparse shape of the same input using the three algorithms (proposed, [2], and [23]), together with the ground truth 3D sparse shape. (View 2) represents the projection of the recovered 3D shapes to the x-y plane. Notice that the results of the proposed method and that of Aldrian et al. [2] are similar.

Figure 5. Side-by-side stem plot of the (a) time (in seconds) needed to recover the 3D sparse shape and (b) 3D shape error (8), for inputs generated from 80 samples of the USF database. The x-axis refers to the sample index of the USF Database. Average time is 0.03, 0.10, and 0.28 seconds and average 3D error is 29.00, 28.79, and 31.80 for the proposed, Aldrian et al., and Zhang et al. approaches, respectively.

Figure 6. Given the input 2D feature points ($x_{inp}$), a camera projection matrix $P$ can be estimated using the 3D mean sparse shape. This camera projection matrix can be used to project all USF database samples to the 2D space.

Figure 7 shows several examples of recovered 3D shapes together with the input images (of unknown pose and illumination) and ground-truth shape. The results are very close, using visual inspection. To quantify the reconstruction accuracy, the author recovers the 3D shape for 80 out-of-training USF samples. The input images are generated with a random pan angle within the range of ($-20^\circ$ to $20^\circ$), where the face moves left-to-right or right-to-left, sideways. Two approaches to recover 3D shape from the input images will be used: (a) Case I - refers to 3D dense shape recovery approach using 2D feature points only and (b) Case II - refers to the 3D shape recovery method from Rara et al. [19], which uses both shading and 2D shape information to perform shape recovery. For each input image, the following measures are used: (a) Height Error - the recovered height map is compared with the ground truth height and
the mean absolute error is computed as

$$\bar{s}_{err} = \frac{1}{N_p} \sum_{i=1}^{N_p} |s_i - s_{GT,i}|$$

where $N_p$ is the number of pixels, $s_i$ and $s_{GT,i}$ are height values at the $i$th pixel position for the recovered shape and the ground-truth shape, respectively, and (b) Surface Orientation Error - the directions of the recovered normal vectors are compared with the ground truth data. The average of the difference angle is computed as

$$\bar{\theta}_{err} = \frac{1}{N_p} \sum_{i=1}^{N_p} \cos^{-1}\left(\frac{\mathbf{n}_i \cdot \mathbf{n}_{GT,i}}{||\mathbf{n}_i|| ||\mathbf{n}_{GT,i}||}\right)$$

where $N_p$ is the number of pixels, $\mathbf{n}_i$ and $\mathbf{n}_{GT,i}$ are normal vectors at the $i$th pixel position for the recovered shape and the ground-truth shape, respectively.

The following are side-by-side visualizations of the mean height (Fig. 8(a)) and surface orientation (Fig. 8(b)) error stem plots, to compare against the proposed 3D shape recovery method using both texture and shape data from [19]. The results show comparable performance, which means that even if shading information is not available, a decent reconstructed 3D shape is still possible.

### 5. Conclusions and Future Work

This paper addressed the problem of estimating 3D shape from an input image in which only 2D shape information is available. Recall that both texture (shading) and shape information were needed for the previous 3D face recovery methods [19][6]. This work showed that both sparse and dense 3D shapes can be estimated from 2D shape in-
formation alone, at an acceptable quality. The paper started with the simpler case of output 3D sparse shapes, where the proposed method was compared with state-of-the-art algorithms, showing decent performance. It was then extended to output 3D dense shapes simply by replacing the sparse model with its dense equivalent, in the regression framework inside the 3D face recovery approach. The next step is to apply this approach to reconstruct faces in FRAD applications similar to [17], where the classical shape-from-X (e.g., stereo, motion and shading) algorithms are not feasible due to input image quality, and use the reconstructed faces to aid in recognition.

References

[1] http://www.cse.usf.edu/~sarkar/, USF DARPA Human-ID 3D Face Database, Courtesy of Prof. Sudeep Sarkar, University of South Florida, Tampa, FL.