3-D Face Recognition Based on Warped Example Faces

Le Zou, Samuel Cheng, Member, IEEE, Zixiang Xiong, Fellow, IEEE, Mi Lu, Senior Member, IEEE, and Kenneth R. Castleman

Abstract—In this paper, we describe a novel 3-D face recognition scheme for 3-D face recognition that can automatically identify faces from range images, and is insensitive to holes, facial expression, and hair. In our scheme, a number of carefully selected range images constitute a set of example faces, and another range image is chosen as a “generic face.” The generic face is then warped to match each of the example faces in the least mean square sense. Each such warp is specified by a vector of displacement values. In feature extraction operation, when a target face image comes in, the generic face is warped to match it. The geometric transformation used in the warping is a linear combination of the example face warping vectors. The coefficients in the linear combination are adjusted to minimize the root mean square error. After the matching process is complete, the coefficients of the composite warp are used as features and passed to a Mahalanobis-distance-based classifier for face recognition. Our technique is tested on a data set containing more than 600 range images. Experimental results in the access-control scenario show the effectiveness of the extracted features.

Index Terms—Digital image processing, face recognition, feature extraction, image classification, 3-D model, 3-D vision.

I. INTRODUCTION

FACE recognition has received broad attention in both academia and industry due to its practical relevance to homeland security. Facial appearance can uniquely identify a person, and it is a primary factor that people use to recognize each other. Although human face images can be affected by illumination, facial expression, makeup, etc., they are noninvasive in nature and easy to be collected in environments where other biometrics (e.g., fingerprint, iris) require cooperation of the subjects. Therefore, face recognition is both an attractive and a challenging area for research. During the past decade, many different 2-D and 3-D face recognition algorithms have been presented [1].

Differentiated by the input image type, these algorithms can generally be categorized into 2-D face recognition and 3-D face recognition. According to the survey by Zhao et al. [1], 2-D face recognition algorithms can be classified into feature-based, holistic, and hybrid approaches. In the holistic category, algorithms use principle component analysis (PCA) [2], linear discrimination analysis (LDA) [3], independent component analysis (ICA) [4], support vector machine (SVM) [5], and neural networks [6] to analyze a face as a whole unit, without distinguishing the individual details (nose, mouth, eyes, etc.). In contrast, algorithms that extract detailed information from these regions are called feature-based approaches [7]–[9]. Hybrid approaches [10]–[14] consider both the overall and detailed information in a face and are regarded as the most promising method [1]. Since 2-D texture pictures can be easily affected by external conditions (viewpoint, lighting, makeup, etc.), 2-D face recognition algorithms are difficult in classifying human faces with high accuracy. Although face recognition methods based on a 3-D morphable model [12]–[14] have made breakthrough progress in successfully solving the problem caused by viewpoint and lighting variation, they rely on manually defined landmark points to initialize the algorithm [12]–[14].

3-D face recognition algorithms, on the other hand, utilize the geometric structure of the face to recognize a person. Since 3-D facial models are invariant to illumination, the impact of lighting variations is moot in this approach. Furthermore, 3-D faces in range images can be rotated and shifted to a standard pose to overcome the problem caused by viewpoint variations. Therefore, this approach becomes increasingly attractive these days. Motivated by early research works [15]–[17] that began about a decade ago, different 3-D face recognition systems [18]–[34] have been proposed. In these schemes, principle PCA [18], [19]; wavelet-signature [20]; curvature [27]; rigid surface matching [22], [28], [30]; iterative closest point (ICP) [21], [23]–[25], [29]; and optimal component analysis [31], [32] techniques are employed in order to achieve good recognition performance.

An important issue with face recognition is the problem of facial expression, which degrades the performance of most current 3-D face recognition systems [35]. While most face recognition systems [20]–[24], [27] do not address the problem explicitly, several schemes have been proposed to handle it [25], [26], [28]–[30]. In the multiple nose region matching method [28], [30] presented by Chang et al., different patches around the nose area in a target face are utilized to compare with those surfaces in the gallery. In their algorithm, these patches are segmented based on the surface curvatures. A target face is identified by measuring the similarity between these patches of the target...
face and those in the gallery. By matching the regions around the nose, which is believed to have relatively low shape variation caused by expressions, their scheme successfully handled the facial expression problem. In the region matching method [29] proposed by Mian et al., different regions of faces in a gallery are segmented and then matched to a target image using ICP. The matching scores are then used to identify the target face. Facial expression problems in target faces are avoided by choosing matching regions that are unlikely to be affected by expression. In the deformable model fitting approach proposed by Lu [25], a deformable model is presented to model the facial expressions and achieves a good recognition rate. On the other hand, Bronstein et al. tried to use a modeling technique to solve the problem [26]. In their proposed 3-D face recognition algorithm [26], generalized multidimensional scaling (GMDS) is used to model facial expressions in order to achieve the expression-invariant performance.

Two other important issues of 3-D face recognition are facial hair and holes on the range images, which have a negative impact on all 3-D face recognition systems [35]. Facial hair in a target face changes the shape of a range image and introduces noises to the face. A hole is an area of missing data in a range image caused by the sensor’s failure of acquiring data from objects. The region matching method [29] explicitly addressed the problem and partially handled the situation by matching the nose and the forehead region only. Their approach, however, does not consider the situation when the forehead region is also partially covered by hair and holes. Although the approach [28], [30] proposed by Chang et al. did not state the problem explicitly, their approach also employs the same strategy of the region matching method [29] and can partially solve the problem.

In this paper, we propose a scheme based on a number of carefully selected 3-D faces called “example faces” and a 3-D face called a “generic face.” The set comprising the example faces is then defined as “example set.” In the design process, the generic face is warped to match each example face. Each such warping can be represented by a displacement vector. In a feature extraction operation, when a target face image comes in, the generic face is warped to match it. The geometric transformation used in the warping is defined by a linear combination of those displacement vectors specified by the warps from the generic face to the example faces. The coefficients in the linear combination are then used as features of the target image and passed to a classifier for recognition.

To illustrate the idea of synthesizing 3-D faces based on a linear combination of warps derived from example faces, consider the 3-D feature space shown in Fig. 1. Here, we have three example faces, each of which corresponds to a particular geometric transformation (shown in Fig. 2), namely, the one that makes the generic face best match that example face. Each axis represents one coefficient in a three-term weighted sum. Each example face lies at 1.0 along its corresponding axis. Further, any point in that 3-D space corresponds to a particular linear combination of those three warps (shown in Fig. 3) and that, in turn, specifies a new face, namely, the one obtained by warping the generic face with the corresponding linear combination of warp parameters. If we restrict the coefficients to sum to unity, then the plane that passes through these three points defines the locus of all faces that can be generated by a linear combination of the three example face warp vectors. Any point on the triangle corresponds to a set of three coefficients that sum to 1.0, and a linear combination of the three corresponding warps, using those weighting coefficients, produces a synthesized face by warping the generic face.

In feature extraction operation, the algorithm adjusts the coefficients to minimize the root-mean-square (rms) error between the target face and the warped generic face. The coefficient vector corresponding to the minimum then constitutes a set of feature values that represents that face.

The feature extraction procedure of our scheme is illustrated in Fig. 4. This procedure starts with an initial coefficient vector \( (w^1_1, \ldots, w^K_K) \), typically all equal to 1/K. Based on this initial
coefficient vector, the generic face is warped to generate a synthesized face $S_I$. Then, the rms error between $S_I$ and the target face $T$ is computed. Using optimization techniques, the coefficient vector of the second iteration $(w_2^1, \ldots, w_2^K)$ can be found. This matching process continues through $M$ iterations until the rms error between the synthesized face $S_M$ and the target face is small enough. After that, the coefficients $(w_M^1, \ldots, w_M^K)$ are used as features to classify the target face $T$. In our scheme, the classification is based on Mahalanobis distance [36].

Compared to the face recognition methods based on the 3-D morphable model [12]–[14], our scheme is different in three ways. First, our scheme utilizes 3-D range images as inputs while the methods based on 3-D morphable model [12]–[14] rely on 2-D texture images. Therefore, our scheme and their approaches belong to two different categories of face recognition systems and have different technical emphasis. In 2-D face recognition, illumination and pose variations cause different appearances of the same human face in 2-D texture images. In order to achieve good recognition performance, it is crucial for 2-D face recognition algorithms to handle the pose and illumination variations. The 3-D morphable model proposed in [12] and [13] and the extended approach [14] presented by L. Zhang et al. focus on handling this problem and have given successful solutions. In contrast, in 3-D face recognition, illumination and pose variations do not affect the shapes of human faces on range images. Furthermore, pose variations can be solved by rotating and shifting human face shapes in range images to a standard pose. Hence, pose and illumination variations are not as much important in 3-D face recognition. In the 3-D face recognition category, holes, facial expressions, and hair may change the overall shape of a 3-D human face on range images, and become important technical issues. Our feature extraction scheme has the ability to extract both overall and detailed information from range images with the presence of holes, facial expression, and hair. Second, our feature extraction scheme is automatic in contrast to the fitting procedure described in [12]–[14], which requires manual initialization. This is due to the fact that the cost function defined in [12]–[14] is based on texture differences and has to be minimized by optimizing both shape and texture parameters of the 3-D morphable model. Therefore, good initial starting values of model parameters are critical and have to be discovered manually. While in our scheme, the cost function is the sum of depth differences and can be minimized by optimizing weighting coefficients only. Without involving of the texture parameters, it is feasible to achieve minimization automatically without falling into local minima. Third, compared to the methods [12]–[14] based on the 3-D morphable model, our feature extraction scheme is more efficient in computation. During the fitting procedure of the 3-D morphable model, in order to avoid occlusions and cast shadows, the 3-D morphable model has to be rendered every 1000 iterations. Due to the large amount of vertices on the 3-D morphable model, this rendering procedure is rather computationally intensive. By contrast, our feature extraction procedure is based on 2-D warping of range images, or in a sense, 2.5-D image transformation. The whole procedure does not require re-rendering of the synthesized face and, thus, our scheme is more efficient in computation. Note that this computation efficiency of our scheme is achieved in the expense of the image acquisition system. Compared to 2-D texture images, 3-D range images may require more complicated acquisition systems.

Compared to other 3-D face recognition algorithms [18]–[23], [26]–[34], our method has several advantages. First, our scheme is insensitive to facial expressions. There are two techniques adopted in our scheme in order to solve the facial expressions problem: 1) matching the facial region that are insensitive to the face expressions and 2) including example faces with facial expressions in the example set. By using these techniques in our scheme, faces belonging to the same person with different facial expressions can be accurately...
differentiated from other people, which has been confirmed by our experimental results. Second, our scheme is robust to holes and facial expressions in range images. In our scheme, holes are adaptively avoided when comparing the differences between a target face and the synthesized face. The facial hair problem, on the other hand, is handled by: 1) including example faces with facial hair in the example set; 2) excluding the regions that can be easily affected by facial hair during the matching procedure; and 3) relying on the whole face region of a range image. By adding examples with hair in the example set, target faces with facial hair can be accurately identified. By matching the face region that is most likely to be unaffected by hair, the effect of facial hair in target faces can be further reduced. Furthermore, by relying on the whole facial region of a range image to identify a person, facial hair has a limited influence on our extracted features and the recognition results. Consequently, our proposed scheme collects all useful information on the target face and successfully solves the problems caused by facial hair and holes. Finally, as a hybrid approach [1], our scheme not only considers the face region as a whole but also the details in target images. By combining the advantages of feature-based and holistic approaches [1], as proven by the experimental results, our scheme can achieve a high recognition rate.

Our scheme shares the same philosophy with the active appearance models (AAM) [37]. Both schemes need manual labeling of landmark points to align images and involve image matching. However, several differences exist. First of all, our scheme is a comprehensive solution for 3-D face recognition comprising both feature extraction and classification, whereas AAM is a general modeling technique designed for studying the statistical nature of images. Second, our scheme is based on 3-D range images while AAM is primarily based on 2-D texture images. Third, the matching procedures in two schemes are different. In our scheme, we achieve image matching by utilizing the stochastic Newton’s method that updates derivative information in each iteration. In AAM, the derivatives are calculated once before the optimization starts.

To summarize, our proposed scheme can be outlined as follows.

1) During the design process, warp a generic model of a face to match each \( K \) example face, retaining the parameters of each \( K \) such warps in a warp vector.

2) In a feature extraction procedure, warp the generic face using a geometric transformation that is a linear combination of the warp vectors that were developed for the example faces.

3) Adjust the coefficients in the weighted sum to minimize the rms error between the input target face image and the warped generic face.

4) Take the coefficients that result in minimum rms error as extracted features that describe the face.

5) Use a linear classifier based on Mahalanobis distance to classify the extracted features.

II. GENERIC FACE WARping

As the first stage of the design process, the goal of generic face warping is to establish point-to-point correspondences between the generic face and example faces. As illustrated previously, we can consider the warped generic face based on a combination of face warps derived from different example faces as a synthesized face. The main thrust is the generic face warping procedure that establishes the point-to-point correspondence between the generic face and example faces. In the sequel, we first describe the landmark points on the example faces and the generic face. Based on these landmark points, the generic face warping procedure that establishes the point-to-point correspondence between the generic face and example faces is introduced.

A. Necessity for Alignment

In order to generate a warped generic face by linearly combining the warps derived from example faces, we need to first develop the proper face warps by comparing between the generic face and example faces. In other words, we need to find out the point-to-point correspondence between the generic face and example faces. Otherwise, when different warps derived from example faces are combined together, points with different physical meanings will be added up, resulting in blurred faces. For instance, if we develop two face warps between the generic face and two example faces [shown as example faces in Fig. 5(a) and (b)] without alignments, a combination of the face warps will generate a blurred face shown in Fig. 5(c). With the alignment, however, pixels with the same physical meaning are aligned together and generate a valid human face, as shown in Fig. 5(d).

B. Landmark Points

In order to find out the point-to-point correspondences between the generic face and example faces, we first manually define 70 landmark points (e.g., eye corners and mouth corners) on each \( K \) example face and the generic face, as shown in Fig. 6. These landmark points all have distinct physical meanings and are easy to be marked manually.\(^1\) Note that the number of landmark points can vary. In our face recognition scheme, we find 70 landmark points are enough to accurately describe the correspondences between the generic face and an example face. The \( x \) and \( y \) coordinates of these 70 landmark points are concatenated to generate position vectors \( \mathbf{X}_L(i) \) and \( \mathbf{Y}_L(i) \), \( i = 1, \ldots, K \). As mentioned previously in Section I, a face in the example set is chosen to be the generic face \( G \). The position vectors of the landmark points on \( G \) are denoted as \( \mathbf{X}_g \) and \( \mathbf{Y}_g \).

\(^1\)Since it is easier to find out the points with distinguishable physical meanings in texture faces, the landmark points are manually marked in the corresponding texture images of the chosen example range face.
With these position vectors, the displacement vectors $\Delta \mathbf{X}_L(i)$ and $\Delta \mathbf{Y}_L(i)$ are defined as

\[
\begin{align*}
\Delta \mathbf{X}_L(i) &= \mathbf{X}_L(i) - \mathbf{X}_g \\
\Delta \mathbf{Y}_L(i) &= \mathbf{Y}_L(i) - \mathbf{Y}_g
\end{align*}
\]

\(i = 1, \ldots, K\). Based on these displacement vectors, we can further generate new displacement vectors

\[
\begin{align*}
\Delta \mathbf{X}_I(W) &= \sum_{i=1}^{K} w_i \cdot \Delta \mathbf{X}_L(i) \\
\Delta \mathbf{Y}_I(W) &= \sum_{i=1}^{K} w_i \cdot \Delta \mathbf{Y}_L(i)
\end{align*}
\]

where $w_i (i = 1, \ldots, K)$ are the weighting coefficients. Depending on $W = (w_1, w_2, \ldots, w_K)^T$, $\Delta \mathbf{X}_I(W)$ and $\Delta \mathbf{Y}_I(W)$ actually represent a set of displacements of the 70 landmark points on the generic face $G$. As we will see in Section II-C, they play an important role in aligning face warps.

### C. Face Warping

In this section, we describe the warping procedure that calculates the displacements of every point on the generic face $G$ based on the displacement vectors $\Delta \mathbf{X}_I(W)$, $\Delta \mathbf{Y}_I(W)$. After this procedure, a point $(x, y)$ in the generic face $G$ will be mapped to a point $(x + \Delta x(W, x, y), y + \Delta y(W, x, y))$ in the warped face $S(W)$. The generic face and example faces used in this procedure are all rendered in range images of size 750 $\times$ 500.

We first divide the generic face into Delaunay triangles [38] based on the 70 landmark points defined by $\mathbf{X}_g$ and $\mathbf{Y}_g$, as shown in Fig. 7. The Delaunay triangulation of a set of points in the plane is a set of triangles connecting the points satisfying an "empty circle" property: the circumcircle of each triangle does not contain any of the points. We employ this triangulation because a fast method for such triangulation is readily available.

The coordinates $\mathbf{X}_g$ and $\mathbf{Y}_g$ of the triangle vertices\(^2\) are then mapped to $\mathbf{X}_I(W) = \mathbf{X}_g + \Delta \mathbf{X}_I(W)$ and $\mathbf{Y}_I(W) = \mathbf{Y}_g + \Delta \mathbf{Y}_I(W)$, respectively.

We can focus on a pair of such corresponding Delaunay triangles as shown in Fig. 8, in which the triangle $T_1$ are mapped to the triangle $T_2$. The vertices $(x_1, y_1)$, $(x_2, y_2)$ and $(x_3, y_3)$ of $T_1$ are then mapped to the vertices $(x'_1(W), y'_1(W))$, $(x'_2(W), y'_2(W))$ and $(x'_3(W), y'_3(W))$ of $T_2$, respectively. Now given a point $(x, y)$ inside $T_1$, we want to find out the corresponding position $(x'(W), y'(W))$ in $T_2$. If we restrict ourselves to linear mappings, we have

\[
\begin{align*}
x'(W) = a(W) \cdot x + b(W) \cdot y + c(W) \\
y'(W) = d(W) \cdot x + e(W) \cdot y + f(W)
\end{align*}
\]

where $a(W) - f(W)$ are the coefficients that need to be determined. Because of the correspondences between vertices of $T_1$ and $T_2$, we have

\[
\begin{align*}
x'_1(W) = a(W) \cdot x_1 + b(W) \cdot y_1 + c(W) \\
x'_2(W) = a(W) \cdot x_2 + b(W) \cdot y_2 + c(W) \\
x'_3(W) = a(W) \cdot x_3 + b(W) \cdot y_3 + c(W)
\end{align*}
\]

and

\[
\begin{align*}
y'_1(W) = d(W) \cdot x_1 + e(W) \cdot y_1 + f(W) \\
y'_2(W) = d(W) \cdot x_2 + e(W) \cdot y_2 + f(W) \\
y'_3(W) = d(W) \cdot x_3 + e(W) \cdot y_3 + f(W)
\end{align*}
\]

From (4) and (5), we can obtain

\[
\begin{align*}
\begin{bmatrix} x'_1(W) \\ x'_2(W) \\ x'_3(W) \end{bmatrix} &= M \begin{bmatrix} a(W) \\ b(W) \\ c(W) \end{bmatrix} \\
\begin{bmatrix} y'_1(W) \\ y'_2(W) \\ y'_3(W) \end{bmatrix} &= M \begin{bmatrix} d(W) \\ e(W) \\ f(W) \end{bmatrix}
\end{align*}
\]

where $M = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}$. Note that $M$ is invertible as long as $\Delta \mathbf{X}_I(W)$ and $\Delta \mathbf{Y}_I(W)$ are linearly independent.

\(\text{Fig. 7. Range face image divided by Delaunay triangles.}\)
as $T_3$ has a nonzero area, so we have

$$\begin{bmatrix} a(W) \\ b(W) \\ c(W) \end{bmatrix} = M^{-1} \begin{bmatrix} x'_1(W) \\ x'_2(W) \\ x'_3(W) \end{bmatrix}$$  \hspace{1cm} (8)$$

and

$$\begin{bmatrix} d(W) \\ e(W) \\ f(W) \end{bmatrix} = M^{-1} \begin{bmatrix} y'_1(W) \\ y'_2(W) \\ y'_3(W) \end{bmatrix}.$$  \hspace{1cm} (9)$$

Let $X'_i(W) = (x'_1(W), x'_2(W), x'_3(W))^T$, $Y'_i(W) = (y'_1(W), y'_2(W), y'_3(W))^T$ and $P = (x, y, 1)^T$, from (8) and (9), we can rewrite (3) as

$$\begin{cases} x'(W) = P'M^{-1}X'_i(W) \\ y'(W) = P'M^{-1}Y'_i(W). \end{cases}$$  \hspace{1cm} (10)$$

Having established the mapping between two arbitrary corresponding triangles $T_1$ and $T_2$, by giving the corresponding vertices of triangles, we can map a point $(x, y)$ in triangle $T_1$ to a point $(x', y')$ in triangle $T_2$. The displacements of the position $(x, y)$ can also be obtained as $\Delta x(W) = x'(W) - x$ and $\Delta y(W) = y'(W) - y$.

If we draw vertical and horizontal lines passing through the landmark points of the generic face $G$, we can further divide $G$ into more than 300 small grids, as shown in Fig. 9. Clearly, the crossing points of these grids are either landmark points themselves or fall into different Delaunay triangles. Displacements of the crossing points that fall into Delaunay triangles can be obtained by using (10). When the displacements of all crossing points of the grids are determined, we can further calculate the displacements of the points inside these grids using bilinear interpolation. Suppose a particular grid has four vertices $(x_i, y_i)$, $i = 1, 2, 3, 4$. Let $\Delta x_i(W)$, $\Delta y_i(W)$, $i = 1, 2, 3, 4$, be the corresponding displacements of the vertices, then the displacement of a pixel with the position $(x, y)$ inside a grid can be calculated as follows by using bilinear interpolation [39]:

$$\Delta x(W, x, y) = (\Delta x_3(W) - \Delta x_1(W)) \frac{y - y_1}{y_4 - y_2} + \frac{\Delta x_2(W) - \Delta x_1(W)}{x_4 - x_1} \frac{x - x_1}{x_4 - x_1} + (\Delta x_1(W) + \Delta x_4(W)) \frac{y - y_1}{y_4 - y_2} + \Delta x_1(W),$$

$$\Delta y(W, x, y) = (\Delta y_3(W) - \Delta y_1(W)) \frac{y - y_1}{y_4 - y_2} + \frac{\Delta y_2(W) - \Delta y_1(W)}{x_4 - x_1} \frac{x - x_1}{x_4 - x_1} + (\Delta y_1(W) + \Delta y_4(W)) \frac{y - y_1}{y_4 - y_2} + \Delta y_1(W).$$  \hspace{1cm} (11)$$

Note that generic face and example faces in the face warping procedure are in range images with size 750×500. We found that face warping procedure can produce satisfying point-to-point correspondence between example faces even with nonrigidities existing (facial expressions, etc.). From the above warping processes, we have achieved the goal of obtaining the displacements of every point on the generic face based on the displacements of landmark points.

After the above warping procedure, the displacement of every point on the generic face $G$ has been calculated, which clearly relies on the weighting coefficients $W$. These displacements can further be considered as displacement images $\Delta X(W, x, y)$ and $\Delta Y(W, x, y)$, which are indexed by a pixel’s position $(x, y)$. Letting the $i$th component $w_i$ of $W$ be one and the rest be zero, we can obtain the displacement images of the $i$th example face (denoted as $E_i$) $\Delta X_i(x, y)$ and $\Delta Y_i(x, y)$, $i = 1, \ldots, K$. With these displacement images, the pixel correspondences between the example faces and the generic face $G$ are established. This accomplishes our goal of seeking the point-to-point correspondences between example faces and the generic face. In Section III, we will use these face warpings derived from example faces to generate synthesized faces using a linear combination.

## III. LINEAR COMBINATION OF FACE WARPINGS

In this section, we will introduce the second stage of the design process. During this stage, face warpings derived from example faces are linearly combined together to generate different synthesized faces based on different weighting coefficients. In addition, we describe the selection process of example faces, which is used to increase the representability of synthesized faces.

### A. Synthesized Faces From a Linear Combination of Face Warpings

The above described warping procedure gives us the displacement images $\Delta X_i(x, y)$ and $\Delta Y_i(x, y)$ ($i = 1, \ldots, K$) of ex-
ample faces. We can further calculate the displacement image of the \(i\)th example face \(E_i^t (i = 1, \ldots, K)\) in the \(Z\) direction (i.e., the depth of the face images)

\[
\Delta Z_i(x, y) = E_i(x + \Delta X_i(x, y), y + \Delta Y_i(x, y)) - G(x, y)
\]

(12)

where \(x\) and \(y\) are the coordinates of the image. From (12), we can reconstruct the example face \(E_i^t (i = 1, \ldots, K)\) as follows:

\[
E_i^t(x, y) = \Delta Z_i(x - \Delta X_i(x, y), y - \Delta Y_i(x, y)) + G(x - \Delta X_i(x, y), y - \Delta Y_i(x, y)).
\]

(13)

Moreover, we can utilize the obtained displacement images of the example images \(\Delta X_i(x, y), \Delta Y_i(x, y)\) and \(\Delta Z_i(x, y)\) \((i = 1, \ldots, K)\) to generate new displacement images

\[
\begin{align*}
\Delta X(W, x, y) &= \sum_{i=1}^{K} w_i \cdot \Delta X_i(x, y) \\
\Delta Y(W, x, y) &= \sum_{i=1}^{K} w_i \cdot \Delta Y_i(x, y) \\
\Delta Z(W, x, y) &= \sum_{i=1}^{K} w_i \cdot \Delta Z_i(x, y)
\end{align*}
\]

(14)

where \(\Delta X(W, x, y), \Delta Y(W, x, y),\) and \(\Delta Z(W, x, y)\) can additionally be used to create a new synthesized face \(S(W)\)

\[
S(W, x + \Delta X(x, y), y + \Delta Y(x, y)) = G(x, y) + \Delta Z(W, x, y).
\]

(15)

Therefore, with the aligned face warps derived from example range faces, we are able to generate new synthesized faces using different weighting coefficients \(w_i, i = 1, \ldots, K\). For instance, if we allocate each displacement image as the same weight \(1/K\), we can generate the average warped generic face, as shown in Fig. 10.

**B. Selection of Example Faces**

Our example set consists of carefully selected range faces, which are denoted as example faces in our scheme. The selection of these example faces is performed in such a way that faces with representative characteristics (for example, faces that are thin, wide, long, short, etc.) will be added. This is because the representability of the synthesized range face depends on the chosen example faces. This can be illustrated in the feature space. Projected into the feature space, the example faces can be considered as points in the feature space. The valid synthesized faces will fall into the “cloud” composed by the example faces, while the unreal synthesized faces will fall out of it, as illustrated in Fig. 11. The larger the “cloud,” the wider range of valid synthesized faces will be generated. Hence, we should select the extreme faces (e.g., long faces and short faces) as example faces, so that the “cloud” composed by the example faces will be as large as possible. Note that once the example faces in the example set are determined, we will not change it during the feature extraction procedure regardless of the input images. Hence, the example set is irrelevant to the training set used in the classification, which will be introduced later.

In our scheme, to find out extreme faces, the PCA technique is applied to analyze a number of range faces in the data set. From Section III-A, we know that for a range face \(E\), we can obtain the corresponding displacement images \(\Delta X, \Delta Y, \) and \(\Delta Z\). These displacement images can be concatenated as a vector \(V\). Let the length of vector \(V\) be \(L\). The purpose of the example face selection procedure is to select \(K\) example faces out of \(H\) range images in the data set. By calculating the mean vector \(\bar{V}\) of these \(H\) vectors \(V_i (i = 1, \ldots, H)\), we can obtain the data matrix \(D = (V_1 - \bar{V}, \ldots, V_H - \bar{V})\). The covariance matrix of \(D\) can therefore be expressed as \(P = (1/H)DD^T\). In order to perform PCA, the eigenvectors \(U_i (i = 1, \ldots, H)\) should be calculated from \(P\). Note that the covariance matrix \(P\) is \(L \times L\), where \(L\) can be a very large number. In order to save calculation complexity, we compute the eigenvectors \(U_i^t (i = 1, \ldots, H)\) from \(P^t = (1/H)DD^T\). Let the corresponding eigenvalue of an eigenvector \(U_i^t\) be \(\lambda_i\). We have

\[
D^T U_i^t = \lambda_i U_i^t.
\]

(16)

By multiplying the matrix \(D\) to the left side of the above equation, the following equation can be obtained:

\[
DD^T U_i^t = \lambda_i U_i^t
\]

(17)

where \(DU_i^t\) is clearly an eigenvector of the covariance matrix \(P\). We can then sort the eigenvectors \((DU_1^t, \ldots, DU_H^t)\) according to their magnitude.
to the values of their corresponding eigenvalues, from large to small. Furthermore, since these eigenvectors may not be normal vectors, it is necessary to normalize them. After the sorting and normalization, we can obtain the eigenvectors \((U_1, \ldots, U_H)\) of \(P\). Each sample vector \(V_i, (i = 1, \ldots, H)\), is thus projected to an eigenvector to obtain its coordinate value \(U_i V_i^T\). Since extreme faces have extreme coordinate values on these eigenvectors (maximum or minimum), based on the coordinate values of the sample vectors and the significance of the corresponding eigenvectors, we are able to choose \(K\) extreme faces from the \(H\) range images in a data set.

In order to reduce the noise in target images caused by facial expressions, we add example faces with facial expressions to the example set, as illustrated in Fig. 12. These example faces with facial expressions can be linearly combined with those example faces with neutral expressions to generate a series of synthesized faces. Depending on the weighting coefficients of the linear combination, different synthesized faces with expressions can be obtained. As shown in Fig. 13, by adding the example face with laughing expression [shown in Fig. 13(e)] to the example with neutral expression [shown in Fig. 13(a)], we can obtain a range of synthesized faces shown in Fig. 13(b)–(d). This is achieved by giving different weighting coefficients to the linear combinations of face warps derived from these two example faces. Therefore, our scheme can produce synthesized faces close to a target face with facial expression and extract reliable features. This is one of the two techniques used in our scheme in order to solve the expression problem. The other technique, as we described in Section IV-B, tries to select matching regions that tend to be unaffected by facial expressions. Note that for illustration purposes, the example faces shown in Fig. 12 are displayed as texture images, although in the example set, the example faces are stored as range images.

After the design process, we can generate different synthesized faces from different linear combinations of face warps by adjusting the weighting coefficients. In Section IV, we introduce the feature extraction procedure that can automatically extract features (weighting coefficients) from a target range image.

### IV. FEATURE EXTRACTION PROCEDURE

Given a target range image \(T\) with a novel face, the goal of the feature extraction procedure is to extract reliable facial features that can differentiate the face from others. Based on the aligned face warps derived from example faces, we design an algorithm to make the warped generic face (synthesized face) as similar as possible to the target face. In this section, we first introduce the cost function that describes the range differences between the synthesized face and a target face. Then, the optimization procedure that minimizes the cost function is presented.

#### A. Cost Function

Suppose we have \(K\) example face \(E_i (i = 1, \ldots, K)\), the generic face \(G\), and a target face \(T\). Given a weighting vector \(W\), let the synthesized range face be \(S(W)\). For a certain position \((x, y)\) on \(G\), the squared range difference between \(S(W)\) and \(T\) becomes

\[
C(W, x, y) = (S(W, x + \Delta X(W, x, y), y + \Delta Y(W, x, y))
- T(x + \Delta X(W, x, y), y + \Delta Y(W, x, y)))^2
\]

\[
= (G(x, y) + \Delta Z(W, x, y)
- T(x + \Delta X(W, x, y), y + \Delta Y(W, x, y)))^2.
\]

(18)

In order to measure the similarity between \(S(W)\) and \(T\), we define the cost function as the sum of the squared difference

\[
C(W, \Omega) = \sum_{(x, y) \in \Omega} (G(x, y) + \Delta Z(W, x, y)
- T(x + \Delta X(W, x, y), y + \Delta Y(W, x, y)))^2
\]

(19)

where \(\Omega\) is a set containing selected points on the generic face. As we will see later, the dynamic selection of points in \(\Omega\) will allow us to avoid holes in target faces. Furthermore, by choosing the points in a region that tend not to be affected by facial hair and facial expressions, our scheme can be robust to facial hair and expressions.

#### B. Matching Region Selection

As just mentioned in Section IV-A, we should dynamically select points in \(\Omega\) so that holes will be avoided in the matching procedure. To understand the adaptive hole avoiding strategy, let us consider picking a point \(P = (x, y)\) the point set \(\Omega\) of the generic face, as illustrated in Fig. 14. Based on the current
weighting coefficients $W$, this point will be compared to the point $P' = (x + \Delta X(W,x,y), y + \Delta Y(W,x,y))$ in the target image. In our scheme, when the gray value of $P'$ in the target image is zero, we believe that $P'$ has fallen into a hole. When that occurs, we simply abandon the point $P$ from $\Omega$. From this selection strategy, the points that fall into holes (with a gray level as zero) will be adaptively avoided in the matching procedure.

In order to avoid the facial hair and facial expression in a target face, we define the matching region $\Psi$ (shown in Fig. 15) in the generic face. In choosing the matching region, the regions that are below the lip and above the eyebrows are discarded to avoid the effect caused by facial hair and facial expressions. During the matching procedure, only points inside the matching region will be selected to the point set $\Omega$. In our scheme, this matching region selection is combined with the techniques mentioned in Section I to achieve reliable feature extraction from target faces.

C. Optimization

After establishing the cost function as the measurement of the similarity between $S(W)$ and $T$, our goal is then to minimize $C(W,\Omega)$ by optimizing $W$. The minimization of the cost function $C(W,\Omega)$ can be achieved by many optimization methods. Here, we choose Newton’s method [13], which is based on the calculation of derivatives of $C(W,\Omega)$, because of its quick convergence. Compared to the optimization procedure in [13], our optimization is automatic. The derivatives of $C(W,\Omega)$ with respect to the $i$th weighting coefficient $w_i$ are given as

$$\frac{\partial C(W,\Omega)}{\partial w_i} = \sum_{(x,y)\in\Omega} 2(G(x,y) + \Delta Z(W,x,y) - T(x + \Delta X(W,x,y), y + \Delta Y(W,x,y)))$$

shown in

$$\frac{\partial^2 C(W,\Omega)}{\partial w_i^2} = \sum_{(x,y)\in\Omega} 2\left(\frac{\partial \Delta Z(W,x,y)}{\partial w_i} - \left(\frac{\partial T(x + \Delta X(W,x,y), y + \Delta Y(W,x,y))}{\partial (x + \Delta X(W,x,y))} \cdot \frac{\partial \Delta X(W,x,y)}{\partial w_i}\right) - \left(\frac{\partial T(x + \Delta X(W,x,y), y + \Delta Y(W,x,y))}{\partial (y + \Delta Y(W,x,y))} \cdot \frac{\partial \Delta Y(W,x,y)}{\partial w_i}\right)\right)^2.$$  

(21)

Beginning with all weighting coefficients being equal to 1/$K$, in each iteration of Newton’s method, we update the next weighting coefficients as follows:

$$W = W - \lambda H^{-1} \nabla C$$  

(22)

where $H^{-1} \approx \text{diag}(1/\partial^2 C(W)/\partial w_i^2)$ ($i = 1, \ldots, K$) is the inverse Hessian matrix and $\lambda \ll 1$ is the learning rate.

We implement two versions of Newton’s method to minimize the cost function. The first version is the standard Newton’s method. We first downsample the generic face $G$ into different subsets based on different scale factors. These subsets are then used as the set $\Omega$ in the optimization. It is obvious that the proper downsampling scale factor is important to achieve the balance of quick convergence and similarity between the synthesized face $S(W)$ and a target face $T$. Let us consider the extreme case of choosing all points in the generic face to form the $\Omega$. When the minimization of the cost function is achieved, we can obtain a synthesized face closest to the target face $T$. However, the convergence could be slow because of the heavy computation in each iteration. On the other hand, if we downsample the generic face using a high scale factor, the size of the set $\Omega$ will be small and the computation will be fast in each iteration. But in this case, even if the cost function $C(W,\Omega)$ is minimized, the actual rms error between $S(W^*)$ and $T$ may not be able to reach the minima.
The second version is a stochastic Newton’s method. In each iteration, a number of points are randomly chosen to the set \( \Omega \) from the matching region on the generic face. So we should be able to determine the number of points in the set \( \Omega \) in each iteration. Here, we face a tradeoff of choosing the proper size of \( \Omega \). On one hand, the smaller size of \( \Omega \) can reduce the computing time of the matching procedure in each iteration. But from (22), we can see that it will also cause the instability of the changing value \( \lambda R^{-1} \nabla C \) in each iteration. In order to smooth the changing of the weighting coefficients, a small learning rate has to be given, resulting in overall slow convergence to the global minima even if computation in each iteration is fast. On the other hand, a larger \( \Omega \) can bring more stability to \( R^{-1} \nabla C \). With more stability, we can give the learning rate \( \lambda \) a larger value, so that even if the computing time in each iteration is longer, the overall matching procedure can be faster. In our scheme, the size of \( \Omega \) and learning rate \( \lambda \) are tested and tuned to reach the best performance.

Both versions of our proposed optimization method have been tested on more than 600 different target images and none of the optimization procedures fall into local minima, which demonstrates the robustness of the proposed optimization method.

After the feature extraction procedure, we obtain the optimized weighting coefficients \( \mathbf{W}^* \), which can be used as features for classification.

V. FEATURE CLASSIFICATION USING MAHALANOBIS DISTANCE

Many classifiers can be used to classify the obtained features from the feature extraction procedure. In our proposed scheme, a linear classifier based on Mahalanobis distance [36] is used, which yields good results as we will see later in Section VI.

A. Feature Preprocessing

The high dimensionality of the features (weighting coefficients) can be redundant; therefore, we again use the PCA to find the dominant dimensions. Let us take a look at the examples on the \( XY \) plane as illustrated in Fig. 16. Using PCA, two orthonormal vectors \( \mathbf{X}' \) and \( \mathbf{Y}' \) can be found. As shown in Fig. 16, these two vectors have the following properties: 1) the covariance of samples is maximized by projecting the samples to the normal vector \( \mathbf{X}' \) and 2) the normal vector \( \mathbf{Y}' \) is orthogonal to \( \mathbf{X}' \). In addition to the two vectors \( \mathbf{X}' \) and \( \mathbf{Y}' \), PCA also produces the corresponding eigenvalues \( \sigma_{\mathbf{X}'} \) and \( \sigma_{\mathbf{Y}'} \), respectively. These eigenvalues represent the variances of the projected samples in the directions defined by \( \mathbf{X}' \) and \( \mathbf{Y}' \). In our example, it is obvious that \( \sigma_{\mathbf{X}'} \) is larger than \( \sigma_{\mathbf{Y}'} \). Therefore, if we discard the vector \( \mathbf{Y}' \) and project all samples to the vector \( \mathbf{X}' \), the dimensionality will be reduced without losing too much information. From this 2-D example, we can generalize the PCA technique to a higher dimensional space. As we will see later in Section VI-D, in our scheme, PCA reduces more than half of the dimensionality of our extracted feature vectors to lower dimensions for analysis.

In addition to reducing the dimensionality of samples, here we use PCA as a preprocessing technique to achieve another goal. As mentioned previously in Section I, our linear classifier is based on Mahalanobis distance. In order to calculate the Mahalanobis distance, the inverse of a data set’s covariance matrix has to be calculated. Therefore, we require the covariance matrix to be invertible. From linear algebra, we learn that a square matrix is invertible if and only if its determinant is not zero. In other words, none of the eigenvalues of an invertible covariance matrix should be zero. Given a \( N \times N \) covariance matrix, from PCA, we can obtain a series of eigenvectors \( (\mathbf{E}_1, \ldots, \mathbf{E}_N) \) sorted by their corresponding eigenvalues \( (\omega_1, \ldots, \omega_N) \), from large to small. If we select the eigenvectors with large eigenvalues and discard the eigenvectors with small eigenvalues, by projecting the samples to the selected eigenvectors, we obtain a new set of samples with nonsingular covariance matrix. This way, PCA can help prevent singular covariance matrices from occurring and ensures accurate calculation of the Mahalanobis distance.

B. Mahalanobis Distance

Given an input feature vector \( \mathbf{x} \), assume we try to assign it to one of the two classes \( \omega_1 \) and \( \omega_2 \), where \( \omega_1 \) represents the class of a particular human face \( P \), and \( \omega_2 \) represents all other sample range images. From the training samples of \( P \), we can calculate the mean vector \( \mathbf{\mu} \) and the covariance matrix \( \Sigma \). As shown in Fig. 17, the Mahalanobis distance from an sample vector \( \mathbf{x} \) to \( P \) can be then expressed as the following:

\[
D_m^2(P, \mathbf{x}) = (\mathbf{x} - \mathbf{\mu})^\top \Sigma^{-1} (\mathbf{x} - \mathbf{\mu}).
\]

(23)

Given a threshold \( h \), we can assign a sample \( \mathbf{x} \) to \( \omega_1 \) if its Mahalanobis distance \( D_m^2(P, \mathbf{x}) \leq h \); otherwise, \( \mathbf{x} \) will be assigned to \( \omega_2 \).

In the classification based on Mahalanobis distance, given the human face \( P \), accurately estimated covariance matrix and mean are the key factors to differentiate \( P \) from other faces.
Hence, sufficient training samples from \( P \) are necessary to be included in the training set. In our scheme, as we will see later, for a human face \( P \), we choose 20 samples to constitute its training set.

VI. EXPERIMENTAL RESULTS

A. Data Acquisition

All range images along with corresponding texture images are obtained from a 3-D camera system. There are a total of 650 range images in our data set containing 113 different human objects. Among these captured range images, about 100 images contain faces with facial hair, expressions, and holes. Each range image is 750 \( \times \) 500 in size. In addition, each 3-D face is normalized by the 3-D camera system beforehand so that the tip of the nose is at the center of the image; and the eyes look straight ahead and they are on a line parallel to the \( x \)-axis. When the pose of a human subject’s face for taking the range image is very different from frontal (e.g., near side pose), the obtained 3-D range images will have many occlusions. Therefore, we require the human subjects taking 3-D range images to cooperate by looking at a mark above the lenses of the 3-D camera system. With this requirement, most occlusions in captured 3-D range images are found to be caused by facial hair.

For illustration purposes, a sample range image with its corresponding texture image are given in Fig. 18. It is seen that the hair in the lady’s face results in holes in the captured images. Our proposed scheme, as proven by our experiments, will be able to tolerate these artifacts.

B. Accuracy and Robustness of the Optimization

In this section, we introduce the experiments that demonstrate the accuracy and robustness of the optimization procedure in our proposed feature extraction scheme. In the experiments, we first measure the accuracy between the positions of landmark points in the synthesized face and target face. Then, the distribution of range differences between the synthesized face and target faces are given. After the optimization, we can use the weighting coefficients \( \mathbf{W}^* \) to calculate the location of the 70 landmark points \( \mathbf{X}_g + \Delta \mathbf{X}_g, \mathbf{Y}_g + \Delta \mathbf{Y}_g \) on the synthesized face. In order to evaluate the accuracy of our feature extraction scheme, we need to compare our results (the location of the landmark points on the synthesized face) with the real location of the landmark points \( (\mathbf{X}_t, \mathbf{Y}_t) \) on the target face \( T \). We find that the landmark points obtained from our scheme are on average within the range of five pixels of the corresponding landmark points on the target images. We give some marking results on the tested target faces shown in Fig. 19 to demonstrate the accuracy of our algorithm. For display purposes, the extracted landmark points are shown on texture images and connected with lines, although our scheme works solely on range images.

The accuracy and robustness of our feature extraction scheme can also be evaluated based on the cost function defined in (19). Let the point set \( \Omega \) be the matching area \( \Psi \) defined in Section IV-B. Assuming there are \( M \) points in the matching area, from (19) we can obtain the average range difference between the target face \( T \) and the synthesized face \( S(\mathbf{W}) \) as

\[
AD(S(\mathbf{W}), T, \Psi) = \frac{1}{M} \sum_{(x,y) \in \Psi} |G(x,y) + \Delta Z(\mathbf{W}, x, y) - T(x + \Delta X(\mathbf{W}, x, y), y + \Delta Y(\mathbf{W}, x, y))|.
\]

After the matching procedure described in Section IV-C, based on the optimized weighting coefficients \( \mathbf{W}^* \), the synthesized face \( S(\mathbf{W}^*) \) can be obtained. The accuracy of the matching procedure can then be evaluated by \( AD(S(\mathbf{W}), T, \Psi) \). Clearly, the smaller the average range difference \( AD(\mathbf{W}^*, T, \Psi) \) is, the higher accuracy the matching procedure achieves. Based on our database that contains more than 600 range images, we test the matching procedure and give the distribution of the average range difference in Fig. 20. It is seen that after the matching procedure, about 95% resulted average range differences are below the 8 range unit, and less than 1% are larger than the 9 range unit, which clearly demonstrates the robustness of our optimization procedure.

C. Feature Extraction Schemes for Comparison

In order to demonstrate the effectiveness of the extracted features (weighting coefficients) generated from our proposed scheme, two other different feature extraction schemes are used for comparison. Similar to the weighting coefficients extracted from our scheme, the feature sets generated from these two schemes are also classified by the Mahalanobis-distance-based classifier to evaluate the performances.

The first chosen feature set is the face profiles, which can be generated directly from the target image \( T \). As illustrated in Fig. 21, when an input range image has been normalized, the
Fig. 20. Distribution of average range difference after the matching procedure.

Fig. 21. Features obtained from the profiles of a target face. (a) The vertical and horizontal lines on a target face for profile extraction. (b) Horizontal profile. (c) Vertical profile.

D. Classification Performance

Our scheme is tested in the access-control scenario, where a group of authorized users is allowed to enter a certain protected area, while the imposters are denied from entering. The groups of authorized users and the imposters are comprised of eight and 105 different human subjects, respectively. Each authorized user has 40 range images captured at different times, while each imposter only has a few images in the database. There are a total of 330 range images from imposters.

The settings of the experiments are shown in Fig. 22. First, range images in the database are processed by the three different feature extraction schemes. After the feature extraction procedure, three different feature sets are obtained. In the three feature sets, a range image will have three different corresponding feature vectors, respectively. A feature vector in the first feature set is comprised of weighting coefficients extracted from our proposed scheme based on warped example faces. The second and third feature sets are from the profiles of 3-D faces and the eigenface method described in Section VI-C, respectively. A feature vector in the feature sets can be considered as a sample of the corresponding range image, and is passed to a Mahalanobis-distance-based classifier for a classification test. In addition to the three feature sets used in classification experiments, for comparison purposes, we also implement the surface matching method presented by X. Lu et al. in [24] and conduct experiments in the same access control scenario. In the surface matching method, the Mahalanobis-distance-based classifier is not used. Instead, the surface difference metric [24] is utilized to perform classification.

In the classification experiments based on the three feature sets, samples belonging to each of the eight authorized users are divided into two halves. For each authorized user, one-half samples are used for training and the other half are left for testing. In the training step, for each of the eight authorized users, based on the training samples, we calculate its covariance matrix and mean, which are then used to construct the Mahalanobis-distance-based classifier. On the other hand, the testing halves of the authorized user samples are combined with the samples from the imposters to form the testing set. During testing, the Mahalanobis distances between an input sample and the eight authorized subjects are calculated. Then, based on a Mahalanobis distance threshold, this testing sample will be classified either as an authorized user or an imposter. By changing the Mahalanobis distance threshold, we can obtain a receiver operating characteristic (ROC) curve for each feature set. To reduce the

Authorized licensed use limited to: Texas A M University. Downloaded on March 10,2020 at 09:06:49 UTC from IEEE Xplore. Restrictions apply.
Fig. 23. ROC curves of various features. (a) ROC curve based on the weighting coefficients. (b) ROC curve based on the profiles. (c) ROC curve based on the eigenface method. (d) ROC curve based on the surface matching method.

effect caused by statistical bias, after one classification experi-
ment, the training halves and the testing halves belonging to the
eight authorized users are exchanged, respectively, to perform
another classification. The final classification results are the av-
erage of the two experiments.

In the experiments based on the surface matching method,
the training set includes eight range images belonging to the
eight different authorized users, respectively. In the matching
procedure, given a testing image, it will be matched to all range
images in the training set. The matching distance is then used as
a metric to make a classification decision. The authorized user
in the range image belonging to the training set with the closest
matching distance to the testing range image will be considered
as identification unless the matching distance is larger than a
threshold. Thus, by changing the threshold, we can also obtain
the ROC curve of the surface matching method. In the surface
matching procedure, given a testing image, about 300 control
points around the eye area are chosen.

In order to evaluate the overall performance, we draw the
ROC curves using the hit rate versus false alarm rates for
our extracted features (weighting coefficients) in Fig. 23(a);
features using profiles in Fig. 23(b); features extracted by
eigenface method in Fig. 23(c); and the surface matching
method in Fig. 23(d). In Fig. 23(a), we can see that the hit rate
reaches 97.2% at the false alarm rate 2%; when the false alarm
rate reaches 5%, the hit rate climbs to 98.5%. Compared to the
performance of weighting coefficients, the features obtained
from the profiles are less efficient in differentiating a person’s
face from others. From Fig. 23(b), we can see that when the
false alarm rate reaches 2%, the hit rate is already 93.4%.
When the false alarm rate is 5%, the hit rate reaches 95.1%.
The performance of using features from the eigenface method
is the worst among the three sets of features. As shown in
Fig. 23(c), when the false alarm rate reaches 2%, the hit rate is
91.2%. At the 5% false alarm rate, the hit rate is about 92.5%.
Thus, from the ROC curves, we can clearly see the advantage
of our extracted features. The reason that features from profiles
have better performance in classification than features from the
eigenface method is because they are less sensitive to the facial
expressions, holes, and hair. Compared to the performances
of the three feature sets using the Mahalanobis-distance-based
classifier, the surface matching method has a lower recognition
rate. However, we need to address that the surface matching
method uses a completely different classification method.
TABLE I

<table>
<thead>
<tr>
<th>Test Schemes</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weighting Coefficients</td>
<td>2.5%</td>
</tr>
<tr>
<td>Profiles</td>
<td>5.1%</td>
</tr>
<tr>
<td>Eigenface method</td>
<td>6.7%</td>
</tr>
<tr>
<td>Surface matching</td>
<td>8.6%</td>
</tr>
</tbody>
</table>

Fig. 24. Misclassified faces (top row) along with corresponding synthesized faces (middle row) and the human subjects to whom the target faces are mistakenly classified (bottom row) when the system operates at the EER point. Note that different classification results can be obtained when the system operates under different configurations.

VII. CONCLUSION

We have presented a novel feature extraction scheme for 3-D face recognition based on warped example faces. We first introduce the face warping procedure based on landmark points defined in generic face. Based on a set of weighting coefficients, a combination of the face warps derived from example faces can be used to warp the generic face as a synthesized range face. After that, the selection of example faces is described. In order to make the synthesized face similar to a target face, Newton’s method is used to minimize the range differences by optimizing the weighting coefficients. After the optimization, the optimized weighting coefficients are used as the feature of the target face. A linear classifier based on Mahalanobis distance is used to verify the identity of the target face. We tested our scheme on a database containing more than 600 range faces. Experimental results showed that the feature extracted from our scheme can accurately differentiate human faces under an access-control setting. Currently, our 3-D face recognition system is implemented in Matlab. On a PC with a 2.4-GHz Pentium processor, the scheme takes about 19 s to recognize a person from its range image. We expect a significant speedup if the scheme is ported in C with optimization.

Our scheme can automatically extract features from a range image captured from a 3-D camera system and is insensitive to holes, facial expressions, and hair. In our scheme, both detailed and the overall geometric information are utilized to recognize a person.

REFERENCES


Mi Lu (S’84–M’86–SM’94) received the M.S. and Ph.D. degrees in electrical engineering from Rice University, Houston, TX, in 1984 and 1987, respectively. In 1987, she joined the Department of Electrical Engineering at Texas A&M University, College Station, where she is currently a Professor. Her research interests include parallel computing, distributed processing, parallel computer architectures and algorithms, computer networks, computer arithmetic, computational geometry, and VLSI algorithms. She has published many technical papers in these areas.

Dr. Lu was Associate editor of the Journal of Computing and Information and the Information Sciences Journal. She was the conference chairperson of the Fifth, Sixth, and Seventh International Conference on Computer Science and Informatics. She served on the panel of the National Science Foundation, the panel of the IEEE Workshop on Imprecise and Approximate Computation, and many conference program committees. She is the Chairperson of about 50 research advisory committees for Ph.D. and Master’s students. She is a Registered Professional Engineer. She has been recognized in Who’s Who in the World (2001, 2003–2006), and Who’s Who in America (2002–2006).

Kenneth R. Castleman received the B.S., M.S., and Ph.D. degrees in electrical and biomedical engineering from the University of Texas, Austin, in 1965, 1967, and 1970, respectively.

He then went to work for NASA’s Jet Propulsion Laboratory, Pasadena, CA. For 15 years, he worked on computer processing of images from space and from Earth-based medical applications. He is the noted author of the seminal textbook Digital Image Processing. He has authored or coauthored more than 60 papers on digital image processing. With more than 30 years of experience, he holds several image analysis and image processing patents and serves on various university and government advisory committees. In 1984, he co-founded Perceptive Systems, Inc. (PSI), Houston, TX, and began commercializing digital imaging technology. PSI designed and built workstations for cytogenetics automation until 2000, when the product line and manufacturing facilities were sold. PSI’s research department, which was headed by him, then became an independent research organization named Advanced Digital Imaging Research (ADIR), League City, TX. In recent years, he has forged ADIR into an industry leader in digital image processing research and development. Under his leadership, PSI and ADIR have developed and deployed many innovative image processing technologies.

Dr. Castleman was inducted into the United States Space Foundation’s Space Technology Hall of Fame in 1994. He is Visiting Committee Chairman of the Department of Electrical and Computer Engineering and Adjunct Professor of Biomedical Engineering at the University of Texas. He is also a member of the Scientific Advisory Board—Center for Light Microscope Imaging and Biotechnology at Carnegie Mellon University, Pittsburgh, PA, and of the Scientific Working Group on Imaging Technology for the Federal Bureau of Investigation (FBI). He has served as a Technical Expert in legal cases ranging from the JFK assassination to bank robberies and patent cases.