Image-Based 3D Face Modeling System

In Kyu Park

School of Information and Communication Engineering, Inha University, 253 Younghyun-dong, Nam-gu,
Incheon 402-751, Korea
Email: pik@ieee.org

Hui Zhang

Computing Lab, Samsung Advanced Institute of Technology, San 14-1, Nongseo-ri, Giheung-eup,
Yongin 449-712, Korea
Email: hui_zhang@sait.samsung.co.kr

Vladimir Vezhnevets

Graphics and Media Lab, Department of Computational Mathematics and Cybernetics,
Moscow State University, Moscow 119899, Russia
Email: vvp@graphics.cs.msu.su

This paper describes an automatic system for 3D face modeling using frontal and profile images taken by an ordinary digital camera. The system consists of four subsystems including frontal feature detection, profile feature detection, shape deformation, and texture generation modules. The frontal and profile feature detection modules automatically extract the facial parts such as the eye, nose, mouth, and ear. The shape deformation module utilizes the detected features to deform the generic head mesh model such that the deformed model coincides with the detected features. A texture is created by combining the facial textures augmented from the input images and the synthesized texture and mapped onto the deformed generic head model. This paper provides a practical system for 3D face modeling, which is highly automated by aggregating, customizing, and optimizing a bunch of individual computer vision algorithms. The experimental results show a highly automated process of modeling, which is sufficiently robust to various imaging conditions. The whole model creation including all the optional manual corrections takes only 2∼3 minutes.

Keywords and phrases: 3D face modeling, image-based system, facial feature detection, generic head model, shape deformation, synthesized texture.

1. INTRODUCTION

The automatic generation of a realistic 3D human head model has been a challenging task in computer vision and computer graphics for many years. Various applications such as virtual reality, computer games, video conferencing, and 3D animation could benefit from convincing human face models. Although there are hardware devices such as laser range scanners and structured-light range finders that can capture accurate 3D shapes of complex objects, they are very expensive and difficult to use. The aim of this paper is to make a convincing human head modeling system that is easy to use by a common PC user. In order to achieve this, it is necessary to use affordable devices such as a digital camera for data acquisition to maximize the possible level of automation along with the controls over the creation process and to develop robust algorithms that generates plausible results even in the case of imperfect input data.

In this paper, we develop a practical automatic system for generating a 3D convincing facial model from frontal and profile images, without imposing strict picture-taking conditions such as illumination, view angle, and camera calibration. The proposed model deformation procedure works well even in the case of imperfect imaging, where the frontal and profile images are not strictly orthogonal. The novelty of the proposed system lies in the development of robust facial feature detection algorithms, which are highly customized and optimized for face modeling purposes. In addition, the system achieves a visually accurate modeling of the ears, which affects the appearance dramatically but has not received a great deal of attention. The overall block diagram of the proposed system is shown in Figure 1.

1.1. Previous works

The existing systems for automatic face model reconstruction can be categorized by the source of 3D facial shape
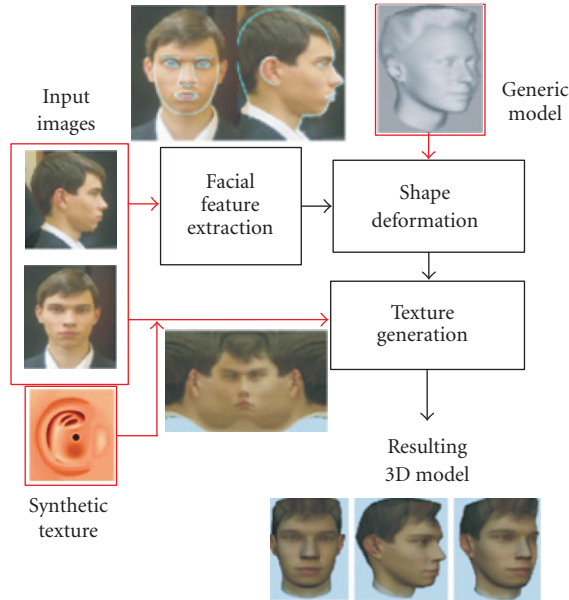
FIGURE 1: Overview of the proposed system.

data they employ. A few researchers have developed techniques to acquire a 3D face directly from a laser scanner or a structured-light range finder [1, 2]. However, it is very expensive and difficult to operate to the extent that the common end-users cannot use it. On the other hand, several studies have been devoted to the creation of face models from two or three views [3, 4, 5]. The views need to be strictly orthogonal, which is difficult to achieve when using a common handheld camera without special mounter setup. A few systems [6] rely on user-specified feature points in several images, which is laborious. Among the other approaches, the methods that utilize optical flow and stereo methods appear to be the farthest step towards the full automatic reconstruction process [7, 8, 9]. However, due to the inherent problems associated with stereo matching, the resulting models are often quite noisy and exhibit unnatural deformations of the facial surface. A few researchers attempted to build a model from just one frontal facial image by using depth estimation from 2D facial feature points [10]. However, the resulting model is yet far from convincing because it lacks both shape and texture information. The statistical model of the human head [11] has been used to introduce additional constraints for face modeling. However, the linear simplified representation is doubtful, and the determination of the coefficients is very difficult to make efficient, robust, accurate, and automatic [12]. A system that makes it possible to combine several head shape data sources (stereo, 2D feature points, silhouette edges) is described in [13]. However, the result was not a complete head and the algorithm is computationally complex.

1.2. Overview of the proposed system

The proposed system focuses on creating a highly automated procedure for a high-quality facial model generation from the frontal and profile images without imposing strict conditions on the picture-taking conditions. Special attention is paid to creating fast and robust facial feature extraction algorithms, which are customized and optimized to face modeling purposes. In addition, considering the limitations of computer vision techniques, this work allows optional user interaction for the manual corrections on top of the automatic reconstruction process. It should be noted that the proposed system does not require any camera calibration, which is in contrast to [13, 14]. In addition, unlike [3, 4, 5], the frontal and profile views do not need to be strictly orthogonal. Note that the input images for this system could be obtained using an ordinary digital camera, which makes the system easy to use and inexpensive.

As shown in Figure 1, the proposed system takes two (frontal and profile) photographs of a person and a generic head model as the inputs to produce a textured VRML model of the person's head. Two types of data are carefully combined: the frontal and profile facial features. The recognition part of the system extracts the facial features robustly. A few algorithms have been developed to detect the individual facial parts including the eye, nose, mouth, ear, chin, and cheek. The generic head model is deformed to coincide with the detected facial features by employing radial-basis functions (RBF) interpolation. The model texture is created by blending the frontal and profile face photos after a color adjustment to give the images a consistent color scale. In addition, a synthetic skin texture is used to fill the model parts, where no texture information can be extracted from the profile and frontal images.

The proposed system is highly modular, allowing developers to replace a particular subsystem with their own algorithm. Different generic head models can be used in the described system without difficulty. The user needs to create a semantics file only in which the correspondences between the facial feature curves and the model vertices are described. Note that it is possible to create a face model from a frontal image only.

1.3. Organization of the paper

The paper is organized as follows. In Sections 2 and 3, the algorithms for the frontal and profile feature detection are described, respectively. Sections 4 and 5 are devoted to the shape deformation and texture generation algorithms, respectively. The experimental results are described in Section 6. Finally, Section 7 provides the conclusive remarks.

2. FRONTAL FACIAL FEATURE EXTRACTION

The frontal feature extraction is comprised of three steps: face region detection, feature position detection, and feature contours (eyes, brows, nose, lips, cheek, and chin) estimation. Each step assumes that the previous ones have been successfully completed and requires a reasonable accuracy of the features detected previously. For this end, user intervention to correct feature detection inaccuracies is possible (but not required) after each step.
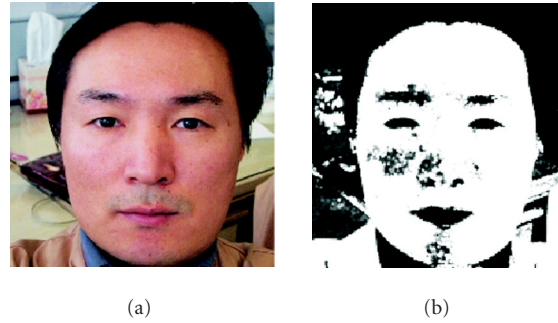
FIGURE 2: Typical skin detection examples. (a) Input image. (b) Detected skin pixels marked in white.

2.1. Face region detection

Many face detection algorithms exploiting different heuristic and appearance-based strategies have been proposed (a comprehensive review is presented in [15]). Among those, color-based face region detection has gained increasing popularity, because it enables the fast localization of potential facial regions and is highly robust to geometric variation of face patterns and illumination conditions (except colored lighting). The success of color-based face detection depends heavily on the accuracy of the skin-color model. In our approach, the Bayes skin probability map (SPM) [16] is employed in the normalized R/G chrominance color space, which has shown a good performance for skin-color modeling. Typical skin detection results are presented in Figure 2.

Skin color alone is usually not enough to detect the potential face regions reliably due to possible inaccuracies of camera color reproduction and the presence of nonface skin-colored objects in the background. Popular methods for skin-colored face region localization are based on connected components analysis [17] and integral projection [18]. Unfortunately, these simple techniques fail to segment the facial region reliably when the face is not well separated from what is mistakenly classified as "skin," as shown in Figure 2b. In order to cope with these problems, a deformable elliptic model is developed, as shown Figure 3a. The model is initialized near the expected face position. For example, the mass center of the largest skin-colored connected region would be a good choice, as shown in Figure 3b. Subsequently, a number (12 in the current implementation) of rectangular probes with an area S_{probe} on the ellipse border deform the model to extract an elliptic region of skin-colored pixels, as shown in Figures 3c and 3d. Let N_{in} and N_{out} be the numbers of skin-colored probe pixels inside and outside the face ellipse, respectively. The density of skin-colored probe pixels inside and outside the face ellipse then control the probe displacement vector $\vec{v}_i$:

$$\vec{v}_i = \begin{cases} l - k_{in} \cdot \vec{n}_i & \text{if } \dfrac{N_{in}}{S_{probe}} < T_1, \\ k_{out} \cdot \vec{n}_i & \text{if } \dfrac{N_{out}}{S_{probe}} > T_2, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where i is the probe index, $\vec{n}_i$ is the probe expansion direction (normal to the current ellipse border and pointing outside), and T_1 and T_2 are the threshold values. Nonnegative coefficient k_{in} and k_{out} control the model deformation speed.

After the displacement vectors for all probes are calculated, an ellipse is fitted to the centers of the repositioned probes. Taking advantage of the elliptical shape of the skin-colored region makes the method more robust, compared with the existing method for skin-pixel cluster detection as reported in [19].

2.2. Eye position detection

Accurate eye detection is quite important in the subsequent feature extraction, because the eyes provide the baseline information about the expected location of the other facial features in the proposed system. Most eye detection methods exploit the observation that the eye regions usually exhibit sharp changes in both the luminance and chrominance, which is in contrast to the surrounding skin. Researchers have used integral projection [20], morphological filters [18], edge-map analysis [21], and non-skin-color area detection [17, 22] to identify the potential eye locations in the facial image. This leads to severe distortion errors when using a nonskin color and low pixel intensities as the eye region characteristics. Luminance edge analysis requires a good facial edge map, which is difficult to obtain when the image is noisy or has a low contrast. However, the detection of sharp changes in the red channel image provides a more stable result, because the iris usually exhibits low red channel values (for both dark and light eyes), compared with the surrounding pixels in the eye white and skin. The proposed method effectively detects the eye-shaped variations of the red channel, which is also easily implemented.

In order to more easily detect a change in intensity, the red channel image intensity is stretched to the maximum range and a variation image is then calculated. Let I be the original red channel image; the pixel value of the variation image at (x, y) can then be defined as

$$V_{n,\alpha}(x, y) = \frac{\alpha}{|R_{n,x,y}|} \sum_{r \in R_{n,x,y}} \left\{ I(r) - \frac{1}{|P_{n,r}|} \sum_{p \in P_{n,r}} I(p) \right\}^2. \quad (2)$$
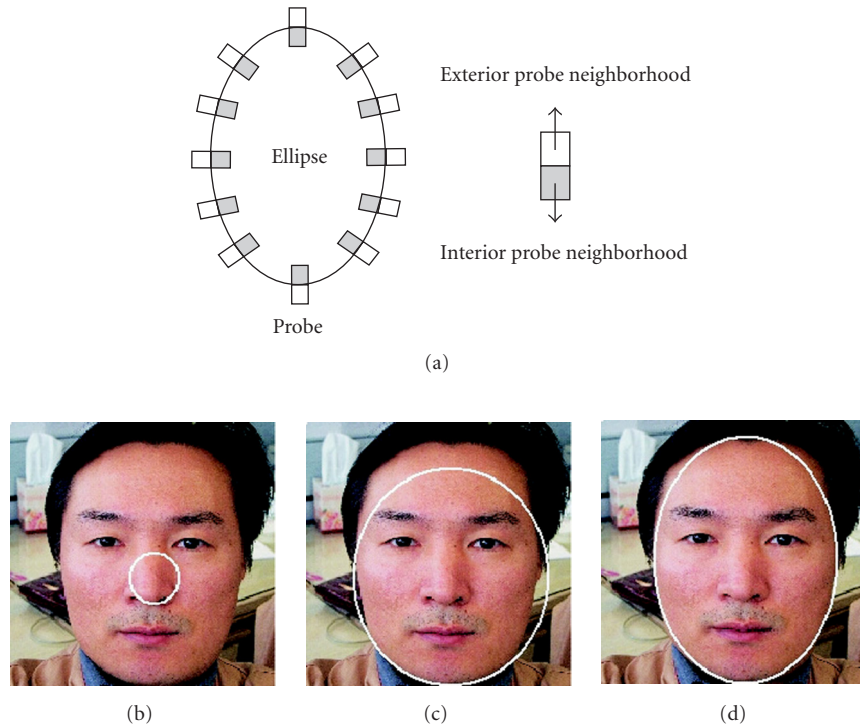
FIGURE 3: Face region detection with deformable model. (a) Deformable model. (b), (c), and (d) Convergence of the deformable model.

Here, $R_{n,x,y}$ denotes an $(n \times 7)$-sized rectangle centered at (x, y), while $P_{n,r}$ is an $(n \times n/3)$-sized ellipse centered at r. The control parameters are the scaling coefficient, α, and the expected size of the eye features, n.

The meaning of the variation image $V_{n,\alpha}(x, y)$ can be described as a dilatation of the high-frequency patterns in the red channel facial image. The variation image is calculated for several (n, α) pairs in order to cope with the high variance of the eye appearance, as shown in Figure 4. This results in a stable and correct behavior for the images with different lighting and quality. The connected components of the pixels with high variation values are then tested to satisfy the shape, size, and symmetry restriction in order to obtain the best-matching eye position for each variation image independently. Finally, different (n, α) configurations are sorted so that the later ones generate a stronger response. Their results are combined in this order so that later results can either provide an output if no response is generated previously, or refine the previous results otherwise.

2.3. Eye contours detection

The eye contour model consists of an upper lid curve in a cubic polynomial, a lower lid curve in a quadratic polynomial, and an iris circle. The iris center and radius are estimated by the algorithm developed by Ahlberg [23]. This is based on the assumptions that the iris is approximately circular and dark against the background, that is, the eye white. Conventional approaches of eyelid contour detection use deformable contour models attracted as a result of the high values of the

luminance edge gradient [24]. Deformable models require the careful formulation of the energy term and good initialization, otherwise an unexpected contour extraction result may be acquired. Moreover, it is undesirable to use luminance edges for contour detection, because eye area may have many outlier edges.

This paper proposes a novel technique that achieves both stability and accuracy. Taking the luminance values along a single horizontal row of an eye image as a scalar function $L_y(x)$, it can be seen that the significant local minima correspond to the eye boundary points, as shown in Figure 5. This observation is valid for many images taken under very different lighting conditions and qualities. The detected candidate pixels of the eye boundary are filtered to remove the outliers before fitting a curve to the upper eyelid points. On the other hand, the lower lid is detected by fitting the eye corners and the lower point of the iris circle with a quadratic curve.

The eyebrows can be detected simply by fitting parabolas to the dark pixels after binarizing the luminance image in the areas above the eye bounding boxes.

2.4. Lip contour detection

In most cases, the lip color differs significantly from that of the skin. Iteratively refined skin and lip color models are used to discriminate the lip pixels from the surrounding skin. The pixels classified as skin at the face detection stage and located inside the face ellipse are used to build a person-specific skin-color histogram. The pixels with low values of a
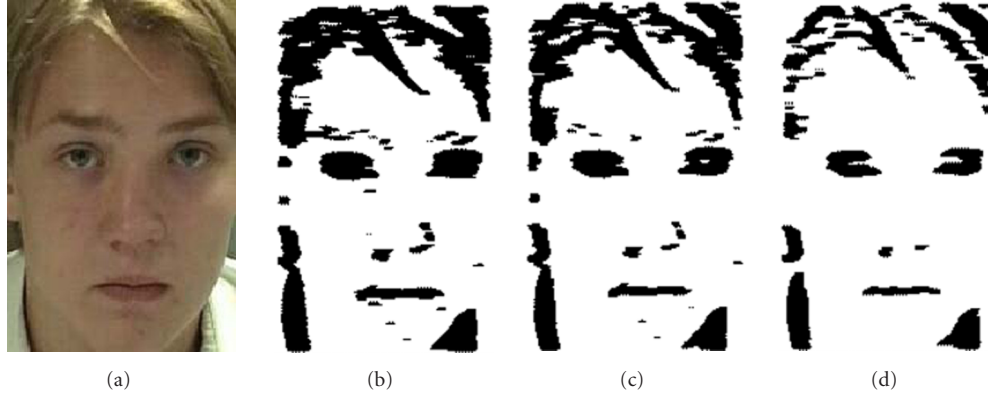
FIGURE 4: Red channel variation images at different scales. (a) Input image with low quality and contrast. (b) $n =$ FaceWidth/30, $\alpha = 1.4$. (c) $n =$ FaceWidth/25, $\alpha = 0.6$. (d) $n =$ FaceWidth/30, $\alpha = 0.6$.
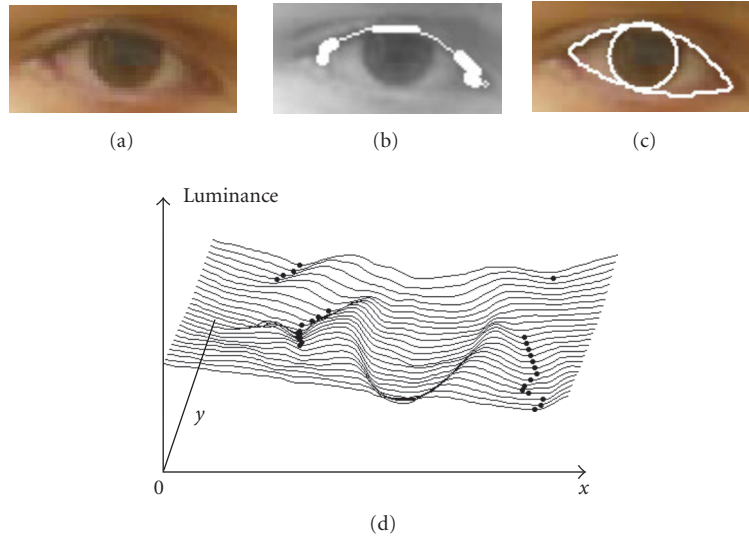

FIGURE 5: Eye contour detection. (a) Source image. (b) Detected eyelid points and fitted curve. (c) Detected contour. (d) Pseudo 3D luminance graph of the eye area. Candidate points for eye border are marked with dark circles.

person-specific skin-color histogram, located at the lower face part, are used to estimate the mouth rectangle. The skin and lip color classes are then modeled using 2D Gaussian probability density functions in (R/G, B/G) color space. It is observed empirically that this color space shows excellence in discriminating lip from skin. The difference between the lip and skin probability for each pixel is used to construct a so-called "lip function image," which is shown in Figures 6b and 6c.

The initial mouth contour is approximated by an ellipse based on the pixel values of the lip function image. The contour points move radially outwards or inwards, depending on the lip function values of the pixels they encounter. Three forces control the displacement of the ith contour point, as given by

$$F_i = F_i^{\text{data}} + F_i^{\text{form}} + F_i^{\text{sm}}, \tag{3}$$

where

$$F_i^{\text{data}} = \begin{cases} -k_{\text{out}} v_c, & f(p_i) \geq T, \\ k_{\text{in}} v_c, & f(p_i) < T, \end{cases}$$

$$F_i^{\text{form}} = k_{\text{form}} \frac{(v_p + v_n) \cdot v_c}{|v_p + v_n|} v_c, \tag{4}$$

$$F_i^{\text{sm}} = 2k_{\text{sm}}(1 + v_p \cdot v_n)\, \text{sgn}\left((p_{i+1} - p_{i-1}) \cdot (v_p^\perp)\right) v_c,$$

where k_{in}, k_{out}, k_{form}, and k_{sm} are positive coefficients that control the contour deformation speed; $f(p_i)$ is the value of the lip function at p_i on the lip contour; T is a threshold; v_n, v_p, and v_c are the unit vectors pointing to the next contour point, the previous contour points, and the ellipse center, from p_i, respectively; $-v_p^\perp$ is a unit vector which is rotated $\pi/2$ clockwise from v_p. The layout of the vectors involved is illustrated in Figure 6a.
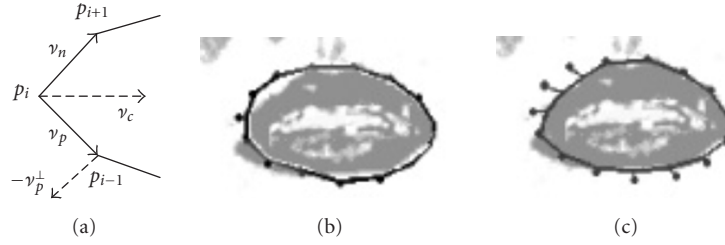
FIGURE 6: Lip contour detection. (a) Layout of control forces. (b) Contour after one iteration. (c) Contour after 12 iterations.
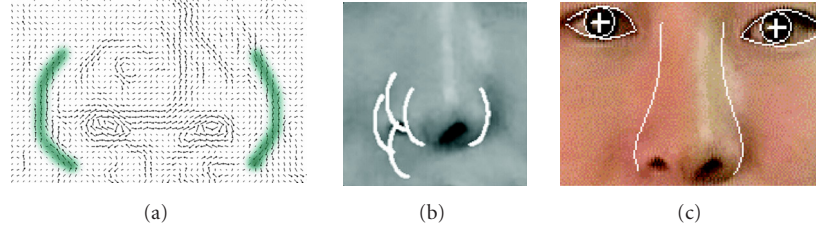

FIGURE 7: Nose contour detection. (a) Luminance gradient vector field of the nose area with nose-side parts marked. (b) Nose-side candidates. (c) Detected nose curves.

In (3) and (4), F_i^{form} is the data force that controls the growth and contraction of the contour. F_i^{form} maintains the contour shape to be close to an ellipse. v_n and v_p are taken as average of several neighbor points in order to make this force affect the global contour form. F_i^{sm} controls the smoothness of the contour, whilst preventing each point from going outside. The force direction is set to be parallel to the ellipse radii $-v_c$, which results in a more stable behavior during convergence. The mouth contour points are moved until their displacements become less than a given threshold. Note that 30 iterations are sufficient in most cases.

In order to match the lip corners exactly, the smoothing force F_i^{sm} is weakened for the points expected to become the lip corners by lowering k_{sm} for the corner neighbor points. The lower part of the mouth contour should be smoother than the upper part, which is also considered in assigning k_{sm}. The intermediate process of contour fitting is illustrated in Figures 6b and 6c.

2.5. Nose contour detection

The representative shape of the nose side has already been exploited in order to increase the robustness, and its matching to the edge and dark pixels has been reported to be successful [24, 25]. However, in cases of a blurry picture or ambient face illumination, it becomes increasingly difficult to utilize the weak edge and brightness information. In our approach, a step further from the naive template methods is made. The proposed approach utilizes the full information of the gradient vector from the edge detector. The nose-side template represents the typical nose-side shape in a fixed scale. In order to compensate for the fixed scale of the template, the nose image (face part between the eye-centers, vertically from the eye bottom to the top of mouth box) is cropped and scaled

to a fixed size. Median filtering is applied before template matching in order to alleviate the noise, whilst preserving the edge information. Luminance gradient vector field is calculated using Prewitt edge mask, which shows less blurring artifacts than Sobel mask when calculating the luminance derivatives. The result of the luminance gradient vector field is shown in Figure 7a. The figure of merit (FoM) at a pixel location q is defined by the following formula:

$$\text{FoM}(q) = \sum_{p \in S(q)} \text{Fit}(p), \tag{5}$$

where

$$\text{Fit}(p) = \begin{cases} 1 & \text{if } \exists r \in \Omega(p) \text{ s.t. } \|\overrightarrow{\nabla I(r)} \cdot \overrightarrow{a(p)}\| \geq 0.9 \\ & \text{and } |\overrightarrow{\nabla I(r)}| > T_1, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

In (6), p, q, and r denote pixel locations; $S(q)$ is the set of template points; $\Omega(p)$ is a 5×5 neighborhood of p; $\overrightarrow{\nabla I(r)}$ is the image spatial luminance gradient vector at r; and $\overrightarrow{a(p)}$ is the normal vector at p on the template curve. T_1 sets the minimum gradient magnitude value to exclude the weak edges. FoM(q) is the counter function that yields the number of pixels in $S(q)$, which exhibit a significant gradient magnitude with a gradient direction close to the template curve normal. Approximately 13% of the maximum figure-of-merit template positions form a set of nose-side candidates, as shown in Figure 7b. A pair of candidates, located with the close vertical coordinates on an approximately even distance from the face central line, is chosen as the most probable nose position, which is also shown in Figure 7b.
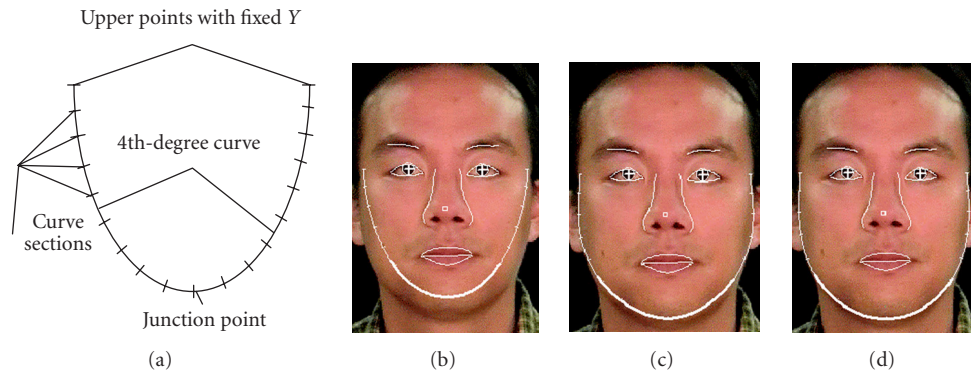
FIGURE 8: Chin and cheek contour detection. (a) Deformable contour model. (b), (c), and (d) The results of contour initialization and fitting.

As shown in Figure 7c, the final nose curves are estimated from the nose-side position and by considering a geometric relation on the general human face structure between the eyes and nose.

2.6.   Chin and cheek contour detection

Deformable models [26] have proven to be efficient tools for detecting the chin and cheek contour [27]. However, in several cases, the edge map, which is the main information source for a face boundary estimation, results in very noisy and incomplete face contour information. A subtle model deformation rule derived from the general knowledge on the human facial structure must be applied for accurate detection [27]. This paper proposes a simpler but robust method that relies on a deformable model, which consists of two fourth-degree polynomial curves linked at the bottom point, as shown in Figure 8a. The model deformation process is designed in such a way that permits the detection of the precise facial boundary in the case of noisy or incomplete face contour. The gradient magnitude and the gradient direction are utilized simultaneously.

The model's initial position is estimated from the already detected facial features, as shown in Figure 8b. After the initialization, the model begins expanding towards the face boundaries, until it encounters strong luminance edges, which are collinear with the model curves. The model curves are divided into several sections. It expands until a sufficient number of pixels occupied by a curve section have an edge magnitude greater than a given threshold and an edge direction collinear with a model curve. The model curves are fitted to the repositioned section points by least-square fitting after evaluating each section's displacement. This is repeated until the model achieves a stable convergence. Figures 8b, 8c, and 8d illustrate this process, showing the convergence to the actual chin and cheek boundary. The lower chin area may exhibit weak edges or no edges at all. In this case, the lower-part sections stop movement when they reach the significant luminance valleys. In order to prevent the excessive downward expansion, the model is not allowed to move lower than the horizontal line, which is derived from the human face proportions.

3.   PROFILE FACIAL FEATURE EXTRACTION

The profile feature extraction consists of two steps: profile fiducial points detection and ear boundary detection.

3.1.   Profile fiducial points detection

The algorithm can be roughly divided into two stages that include profile curve detection and fiducial points detection. The robustness is achieved by a feedback strategy: the later steps will examine the results from previous steps. When erroneous intermediate results are detected, the algorithm will automatically go back to previous steps and fix the errors with additional information.

3.1.1.   Profile curve detection

Profile facial curve is detected as the boundary of the face region, which is the largest skin-colored connected component, or another near the image center when the two largest components have a comparable size. Note that the same skin-color classification algorithm for the frontal image is employed.

However, the color-based algorithm gives incorrect results in many cases, as shown in Figure 9. For example, the detected face region will not include the nose when a strong shadow separates the nose from the face, as shown in Figure 9b. This failure case can be recognized by searching for the nose candidate, which is the skin-colored connected component near the face region in the horizontal direction. The nose tip is detected as an extreme point along the horizontal direction (it is the right-most point in this implementation) in the currently detected face region (or the nose region when the nose is separated from the face). In addition, the nose bridge is estimated by fitting a line segment to the local edge points, beginning from the nose tip. The next stage is to check if there is another failure condition, that is, the incomplete nose area due to strong illumination, as shown in Figure 9e. This can be carried out by calculating the distance from the nose bridge to the face region boundary points. After the failure cases of skin-color segmentation are recognized, a pyramid-based area segmentation algorithm [28] in the Intel Open Source Computer Vision Library (http://www.intel.com/research/mrl/research/opencv)
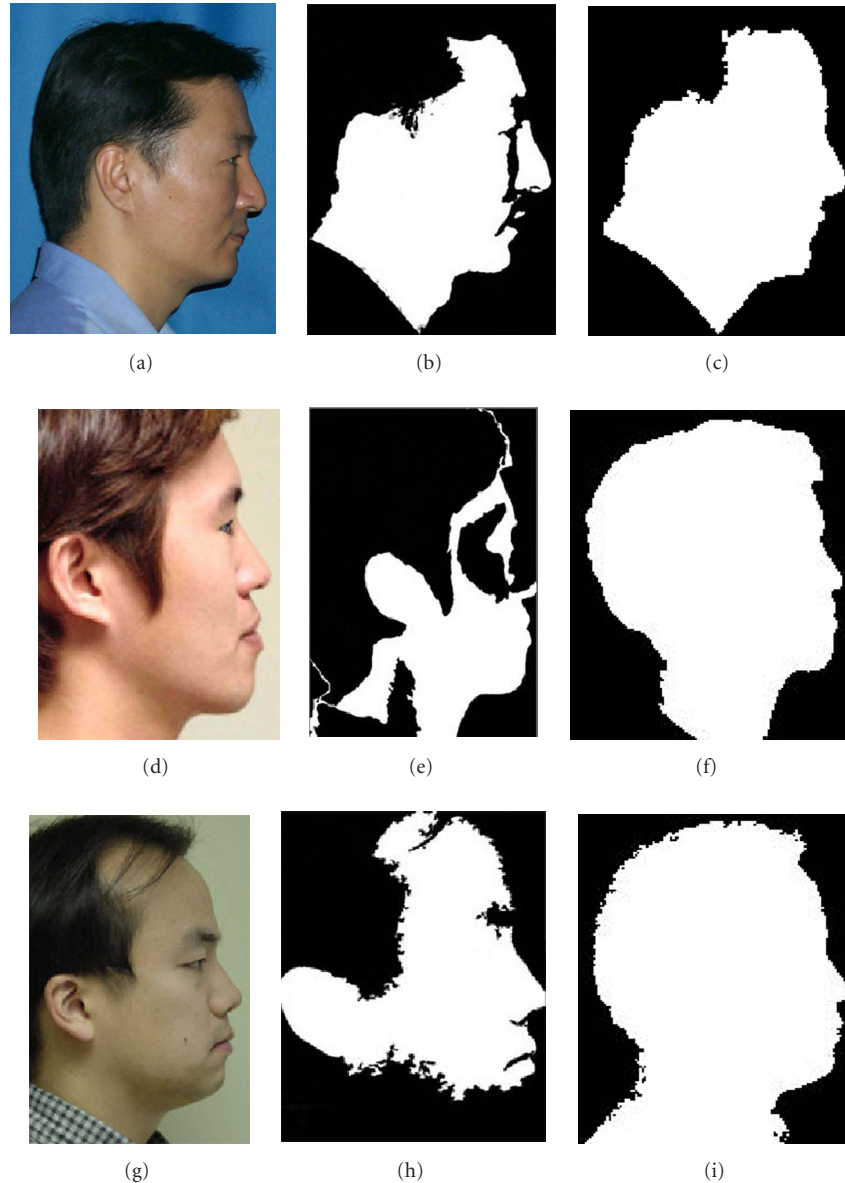
FIGURE 9: Failure cases of color-based segmentation and the corrected results by area segmentation. (a), (b), and (c) The case of separate skin and nose. (d), (e), and (f) The case of incomplete nose area. (g), (h), and (i) The case of incomplete chin area. (a), (d), and (g) Input images. (b), (e), and (h) The results of color-based segmentation. (c), (f), and (i) The results of area segmentation.

is used to detect the profile curve, as shown in Figures 9c, 9f, and 9i.

3.1.2. Fiducial points detection

The profile fiducial points are detected after extracting the profile curve. They are defined as a few characteristic points positioned on the profile curve. First, a "profile function," $x = x(y)$ is constructed, where y varies along the vertical direction, and x denotes the rightmost x coordinate. It is also smoothed with 1D Gaussian filter to eliminate noise. Figure 10 shows a typical profile function, where a few fiducial points to be detected are also marked. The "nose tip" (A in Figure 10) is the global maximum position. The "nose

bridge top" (B in Figure 10) is the ending point of a line segment representing the nose bridge. A search area for detecting the "under-nose point" is then defined (C in Figure 10) based on the length of the nose bridge, and the position where the curvature is maximized in this area is believed to be the detection result. Afterwards the profile curve below the "under-nose point" is approximated as a polyline with an adaptive polyline fitting algorithm. The "chin point" (D in Figure 10) is the first polyline vertex whose successive line segment direction is close to the horizontal direction; the "neck top point" (E in Figure 10) is the first polyline vertex after the chin point whose successive line segment direction is close to the vertical direction. Finally, the

FIGURE 10: Typical profile function. Nose tip (A), nose bridge top (B), under-nose point (C), chin point (D), and neck top point (E) are marked.

lip points are positioned as the interpolation results of the under-nose point and the neck top point with predetermined coefficients. The distance from the nose tip to the chin point is then calculated. If it is too short, the third failure condition of the skin-color segmentation is recognized, as shown in Figure 9h. This is due to the insufficient illumination. For this case, the algorithm will return to the previous stage in order to redetect the profile curve with the area segmentation algorithm.

3.2. Ear boundary detection

Since the local low-level clues are usually weak and erroneous in the area around the ear, a curved template, which represents the priori knowledge about the human ear, is utilized to detect the ear boundary. The complete ear detection consists of three steps: (1) profile image normalization to compensate for the different scale and orientation, (2) ear initialization to match the template with the image ear and to translate it to an initial position, and (3) ear refinement to deform the template in order to match the accurate ear boundary.

3.2.1. Profile image normalization

Two profile fiducial points, the nose bridge top and the chin point, are selected as the calibration points for normalization. The original image is rotated to make the segment connecting them vertical, which is then scaled to make the distance between them a predefined value. These two points are selected because they could be detected robustly, and they are distant enough so that the normalization is less sensitive to their detection errors. In this stage, a rectangle is also defined as the search area for the ear by statistical analysis on the relative positions between the ears and the calibration points in our test images.

3.2.2. Ear initialization

As described previously, a curve template is utilized for the ear initialization and refinement. In this implementation, the template is a five-degree polynomial, as shown in Figure 11 (the thick white curve). The "skin-color boundary" (the boundary of the face region detected with the skin-color classification) is used for ear initialization because it coincides

with the ear boundary at some partial segment in most cases, as shown in Figure 11. The problem is then to identify the corresponding partial segment between the template and the skin-color boundary inside the search area. After the scale and orientation normalization, it can be solved with a simple curve-matching algorithm based on the similarity of the curve tangent.

In more detail, the two curves are preprocessed to be 4-connected, thereby avoiding local duplication. The resultant point sets are denoted as $\{\mathbf{q}_i \in \mathbf{R}^2\}|_{1 \leq i \leq N}$ for the template, and $\{\mathbf{p}_j \in \mathbf{R}^2\}|_{1 \leq j \leq M}$ for the skin-color boundary, respectively. Next, two displacement arrays are constructed as $\{\mathbf{VQ}_s = \mathbf{q}_{a(s+1)} - \mathbf{q}_{as}\}$ and $\{\mathbf{VP}_t = \mathbf{p}_{a(t+1)} - \mathbf{p}_{at}\}$, where a is a coefficient for the sampling step. $l(s, t)$ is evaluated as the maximum integer l that satisfies

$$\sum_{m=1}^{l} ||\mathbf{VQ}_{s+m} - \mathbf{VP}_{t+m}|| \leq \delta, \tag{7}$$

where δ is a threshold to measure the similarity in the tangential direction at $\mathbf{q}_{as}$ and $\mathbf{p}_{at}$. The position (s_m, t_m), where $l(s, t)$ is a maximum, gives a match result as $\{\mathbf{q}_i\}|_{as_m \leq i \leq a(s_m + l_m)}$ and $\{\mathbf{p}_j\}|_{at_m \leq j \leq a(t_m + l_m)}$. Finally, the template is translated based on the partial segment match. The proposed initialization method works very well in experiments, as shown in Figure 11.

However, the skin-color boundary has no coincidence with the ear boundary when hair is too short or the illumination is too strong, as shown in Figure 12. Such cases can be automatically detected during the previous match process by selecting a threshold for $l(s_m, t_m)$. In this case, an attempt is made to initialize the ear by employing edge information. A Nevatia-Babu edge detector [29] is selected in our implementation for its simplicity and good performance. The edges are thinned and linked together to generate long segments. The previous curve-matching algorithm is then utilized again between the template and the edge segments in order to obtain the matched partial segments and a translation vector for the template. In order to determine the "best" edge segment, the following factors are considered.

(i) The length of the matched partial segment for each edge segment: longer is better.

(ii) The translated template position: it should be inside the search area.

The result of ear initialization is shown in Figure 12.

3.2.3. Ear refinement

Based on the initialized ear template and the matched segment on the ear boundary image, a contour-following method is developed to deform the template to match with the whole ear boundary.

In more detail, the template with line segments is approximated using an adaptive polyline fitting algorithm. The first line segment that has its vertex on the ear boundary is selected as the starting position of contour following. This vertex is denoted as Cont_n, and the next vertex along the polyline
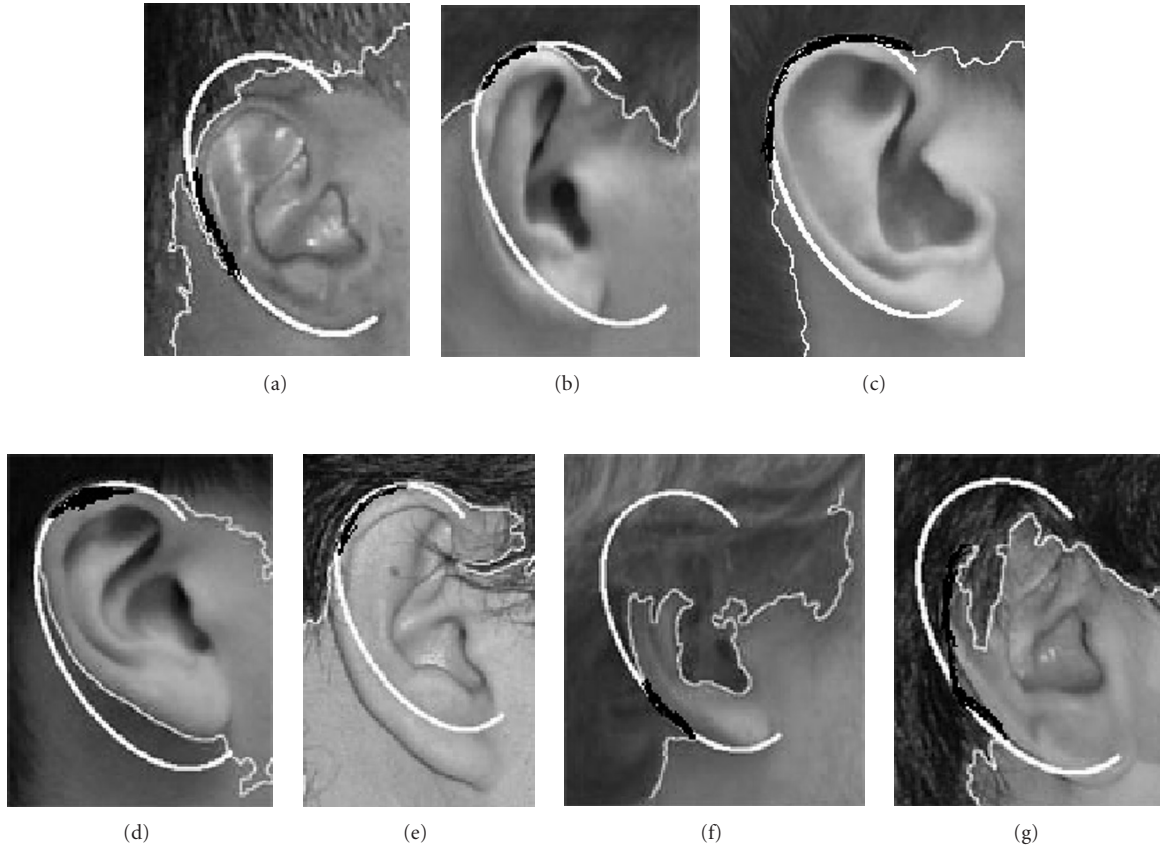
FIGURE 11: The results of ear initialization with skin-color boundary (thin grey curve), translated ear template (thick white curve), and the matched partial segments (thick black curve) on both skin-color boundary and the ear template.
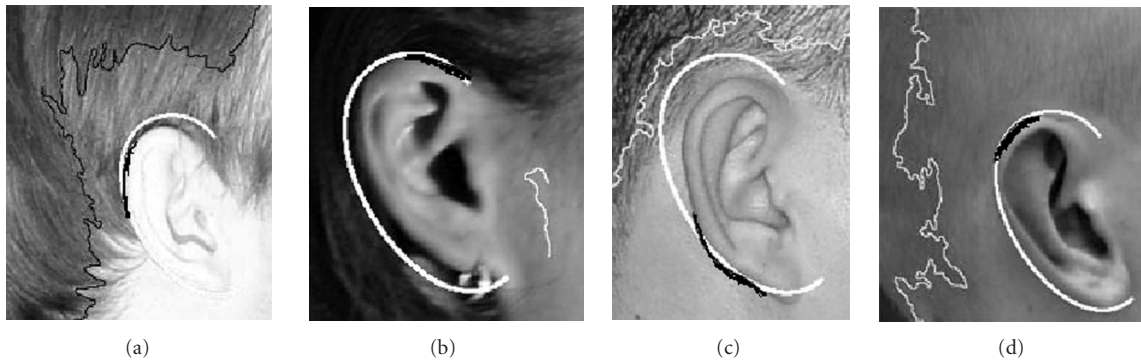

FIGURE 12: The results of ear initialization using edge segments when skin-color boundary is incorrect.

is denoted as $Cont_{next}$. The segment is rotated to a new position that gives the best match evaluation, which will be defined in the next paragraph. All the connected segments after $Cont_n$ are rotated together, as illustrated in Figure 13. Finally, letting n = next, this operation is performed iteratively to deform the whole template. This procedure is employed twice for both directions of the polyline.

The match evaluation between a particular segment and the image is defined by combining two factors. One is the local edge strength, which is evaluated as the sum of the edge gradient magnitudes along the segment. The pixels inside the ear are required to be brighter than the neighboring skin pixels. The other is the segment similarity, that is, the sum of the intensity values of all the segment pixels in a line segment
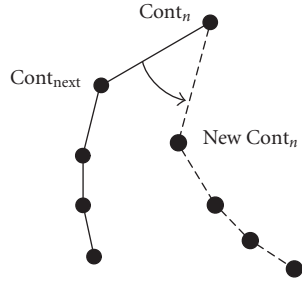
FIGURE 13: The template segments before (solid line) and after (dashed line) rotation, during ear contour following.

should not be too different from the value computed for its precedent. If any of these two requirements is not satisfied, the contour following terminates with a failure alarm. The initialized ear template is used for the failure cases. Note that this occurs mainly when the hair occludes ear.

For the ear bottom direction, the template length may be not adequate for matching with the ear boundary. Therefore, the "tail" of the template is replaced with a long line segment. The real bottom position of the ear is found by examining the edge strength along the deformed template.

4. SHAPE DEFORMATION

In the proposed system, new models are created by deforming the generic head model. The generic model is created by averaging a few laser-scanned faces and is simplified to approximately 20 000 triangles. The generic model is deformed to fit the extracted facial features using the radial-basis functions (RBFs). Two algorithms are developed in order to cope with the conflicting facial features in the frontal and profile images. One is the preprocessing of the detected profile facial features; the other is the new deformation algorithm.

4.1. RBF interpolation

Suppose there are two corresponding data sets, $\{\overline{\mathbf{u}}_i \subset \mathbf{R}^3\}$ and $\{\overline{\mathbf{u}}_i' \subset \mathbf{R}^3\}$, representing N pairs of vertices, that is, the key points, before and after the data interpolation. The following $f(\mathbf{p})$ can be determined as the deformation function of the whole 3D space [30]:

$$f(\mathbf{p}) = \overline{\mathbf{c}}_0 + [\overline{\mathbf{c}}_1 \overline{\mathbf{c}}_2 \overline{\mathbf{c}}_3]\mathbf{p} + \sum_{i=1}^{N} \overline{\Lambda}_i \varphi_i(|\mathbf{p} - \overline{\mathbf{u}}_i|), \qquad (8)$$

where $\mathbf{p}$ is any 3D position and φ_i is the RBF for $\overline{\mathbf{u}}_i$. In this implementation, $\varphi_i(r) = e^{-r/K_i}$, where K_i is a predetermined coefficient that defines the influencing range of $\overline{\mathbf{u}}_i$. $\overline{\mathbf{c}}_0$, $\overline{\mathbf{c}}_1$, $\overline{\mathbf{c}}_2$, $\overline{\mathbf{c}}_3$, and $\overline{\Lambda}_i$ are all vector coefficients, which are determined by solving the following constraints:

$$f(\overline{\mathbf{u}}_i) = \overline{\mathbf{u}}_i' \mid_{i=1}^{N}, \qquad (9)$$

$$\sum_{i=1}^{N} \overline{\Lambda}_i = 0, \qquad (10)$$

$$\sum_{i=1}^{N} \overline{\mathbf{u}}_{i,j} \overline{\Lambda}_i = 0 \mid_{j=x,y,z} . \qquad (11)$$

Based on the RBF-based deformation framework and the 3D correspondences at the given key points between the features in the generic model and the input image, the generic model is deformed in order to generate a complete individual model by feeding all the vertices in the generic model into (8). The key points are defined as the fiducial facial features, as shown in Figure 14.

4.2. 2D/3D key points generation

The key points represent perceptually important facial feature points, by which most of the geometric characteristics of individual humans can be determined. In this approach, 160 fiducial points are sampled from the detected facial features. The key points are sampled along each contour such that it has the same sampling rate in the relative contour length as that of the key points on the corresponding contour of the 3D generic model. Note that the key points in the 3D generic model are predefined.

4.3. Preprocessing of the profile key points

Before performing the model deformation, the detected profile key points are preprocessed in order to cope with the incoherent frontal and profile facial features due to the nonorthogonal picture-taking condition. Two steps, a global transformation and a local one, are followed in order to complete this task.

During the global transformation, all the key profile points are scaled and rotated to match the frontal image. Scaling is necessary to compensate for the different focal length and viewing distance. The scale coefficient is determined, based on the vertical distance of the two fiducial points, that is, the nose bridge top (NBT) and the chin point (CP). Rotation is also needed because the camera may be slightly rotated around the optical axis, or the pitch angle of the specific head is changed whilst taking the picture. However, it is difficult to determine the rotation angle because of the lack of available information. In this system, the angle between the vertical axis and the line connecting the NBT and the CP is assumed to be uniform for individuals and an attempt is made to rotate the profile key points such that the angle between the line and the vertical axis is equal to the angle observed in the 3D generic model.

During the local transformation, the profile line is partitioned into different segments based on a few key points such as the nose, mouth, and chin points. Each segment is scaled separately in the vertical direction to coincide with the frontal features, as shown in Figure 15.

4.4. Model deformation

A three-step deformation procedure is developed in order to deform the generic model and generate a specific 3D face model, which is described as follows. First, RBF interpolation is applied with the frontal key points. Assume that the generic model faces the $\mathbf{Z}$ direction, while the horizontal and vertical directions of the frontal face are aligned to the $\mathbf{X}$ and $\mathbf{Y}$ coordinates, respectively. The $\mathbf{X}$ and $\mathbf{Y}$ coordinates of the displaced key points ($\overline{\mathbf{u}}_i'$ in (9)) are set to their corresponding
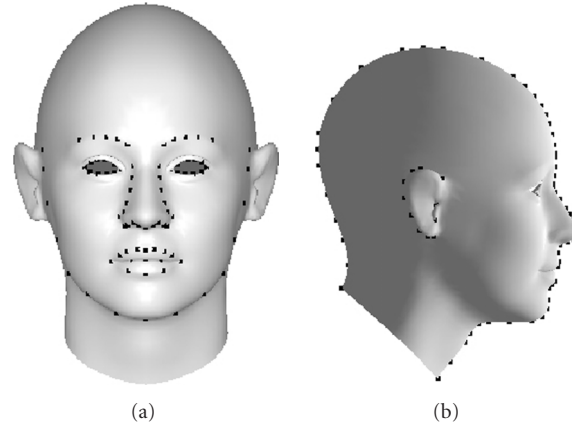
FIGURE 14: The key points marked in the generic model. (a) Frontal key points. (b) Profile key points.
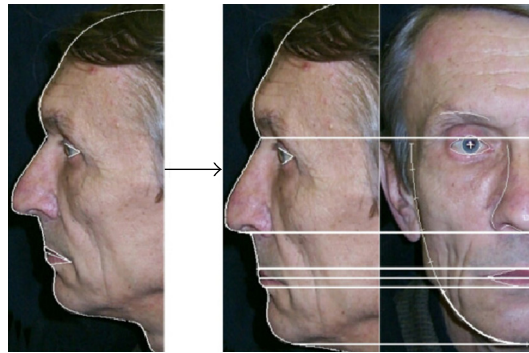

FIGURE 15: Local transformation on profile feature points.

image positions under the frontal view, and the **Z** coordinate remains the same as in the generic model. After RBF interpolation, the accurate match between the model and the frontal feature is achieved. The deformation result of this step is denoted as M1.

Next, the profile key points are used to perform the RBF interpolation. The **Y** and **Z** coordinates of the displaced key points ($\overline{\mathbf{u}}'_i$ in (9)) are set to their corresponding profile image positions, and the **X** coordinates remain the same as in the generic model. During this step, the **Z** values are determined for all model vertices. The deformation result of this step is denoted as M2.

Finally, all the key points are used to determine the resulting head shape. In this step, the profile key points retain their positions in M2. Regarding the frontal key points, their **Z** coordinates are set according to M2, and the **X** and **Y** coordinates are set according to M1. The final shape matches the detected 2D facial features in both view angles. Figure 16 shows a few created head shapes.

5. TEXTURE GENERATION

Being mapped onto the deformed 3D wireframe model, the texture provides the major contribution to the visual appearance and the perception of the model. This section describes

the texture generation method, including the frontal and profile facial color adjustment, texture mapping to a public plane, and blending the frontal and profile textures.

5.1. Color adjustment

The color scale of the skin pixels in the frontal and profile images may be different due to the difference in the lighting condition or the camera color balance. This produces undesirable artifacts in the blended texture. In order to avoid this, color transfer between frontal and profile images is performed in the $l\alpha\beta$ color space [31]. Note that the $l\alpha\beta$ color space is known to have a low correlation between the color channels whilst having a closeness to the channel meanings in the human color perception. The results of the color adjustment are shown in Figure 17, in which source image, target image prior to adjustment, and the target image after adjustment are shown in each row, respectively.

5.2. Texture mapping to a public plane

In order to combine the texture from different view angles, a public UV plane containing the texture coordinates of the model vertices is created. This is a 2D plane where the points are matched with the vertex positions of the generic model. The plane has normalized coordinate space, that is, $\mathbf{M} = \{(u, v) \mid u, v \in [0, 1]\}$. Such a public coordinate space
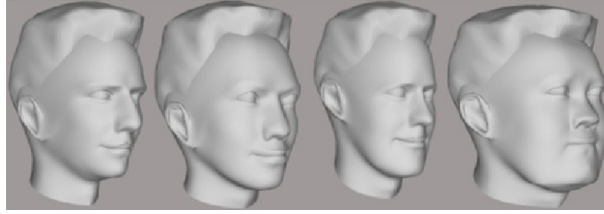
FIGURE 16: Examples of the deformed head models.

(a)   (b)   (c)

(d)   (e)   (f)
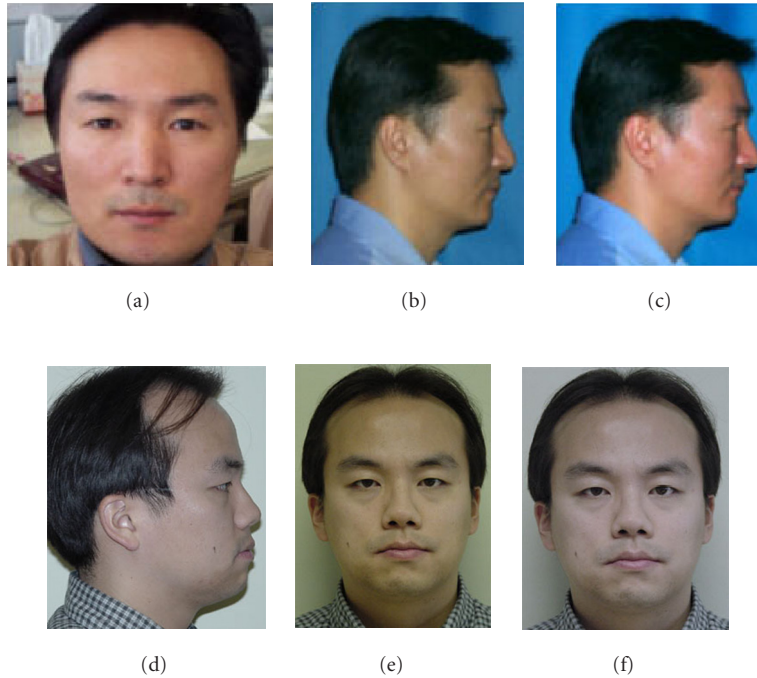
FIGURE 17: The results of color adjustment. (a) and (d) Reference image. (b) and (e) Input source image before adjustment. (c) and (f) Output image after adjustment.

also makes it easy to combine the texture from a real photo with a synthetic texture.

Because the human head is similar to a sphere-like shape, spherical texture mapping is used to ensure the even distribution of the model vertices across the texture coordinate space. The directly generated space is further repaired manually in order to solve the overlap problem, (i.e., in ear areas) on the UV plane. Note that this procedure requires no user interaction, because the UV plane is created with the generic model and remains constant when the individual models are created.

In order to create a textured model, the color-adjusted photos are mapped onto a UV plane. Since the model key points have been fitted to the image feature points, the correspondence between the model vertices and the image feature positions are already known. The mapping is used twice for the frontal and profile photos, resulting in two individual texture images.

5.3. Texture blending

The frontal and profile textures should be blended together to generate a final texture on a UV plane, which is then mapped to the created 3D model during rendering. For each point $p \in \{(x, y) \mid x, y \in [0, 1]\}$ in the UV plane, the color of the blended texture is obtained by the interpolation as follows:

$$C(p) = k_f(p)C_f(p) + k_s(p)C_s(p), \tag{12}$$

where C_f, C_s are the colors at point p for the frontal and profile textures; k_f, k_s are the normalized weights for the different textures, yielding $k_f(p) + k_s(p) = 1$ for every p.

The weights of the texture blending are generated by a multiresolution spline algorithm [32]. Based on a Gaussian pyramid decomposition, this image blending algorithm can achieve a smooth transition between images without blurring or degrading the finer image details. This also benefits
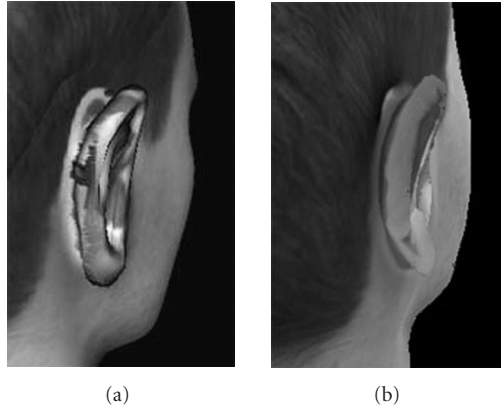
FIGURE 18: The effect of using synthetic texture. (a) Simple texture mapping. (b) Improved result by using the public UV plane and the synthetic texture mapping.
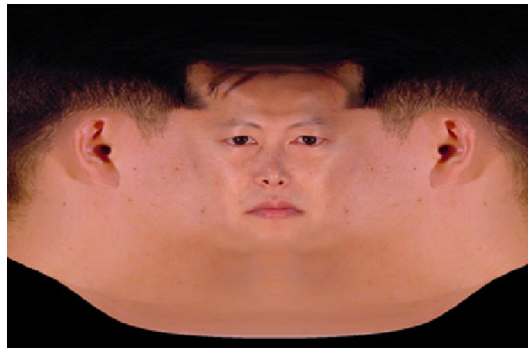

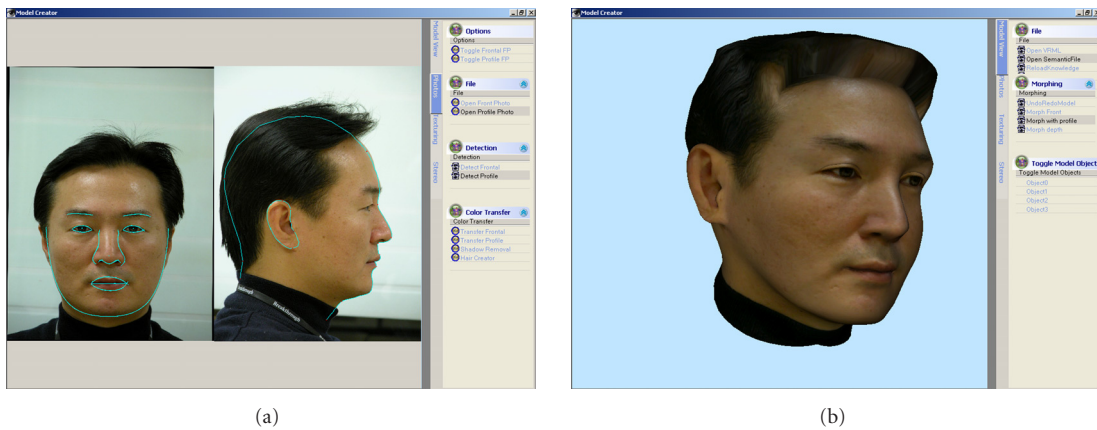FIGURE 19: Generated combined texture on the public UV plane.

FIGURE 20: Developed face modeling system. (a) Feature extraction module is working. (b) The generated 3D model is being rendered.

the utilization of the blending boundary with an arbitrary shape.

In this approach, a couple of additional techniques are used to improve the texture quality. One is the utilization of the synthetic texture to improve the texture around the ear area, where a synthetic texture is assigned to the occluded area. The improved ear texture is shown in Figure 18. The other is to use some artificial pixels for the neck area, where a strong shadow and self-occlusion are often observed.

The artificial pixel colors are randomly generated according to the individual skin-color statistics. Figure 19 shows a typical resulting texture after these three steps.

6. EXPERIMENTAL RESULTS

Figures 20a and 20b shows two screen shots of the proposed face modeling system, where the detected features and the textured 3D models are shown, respectively. The system is
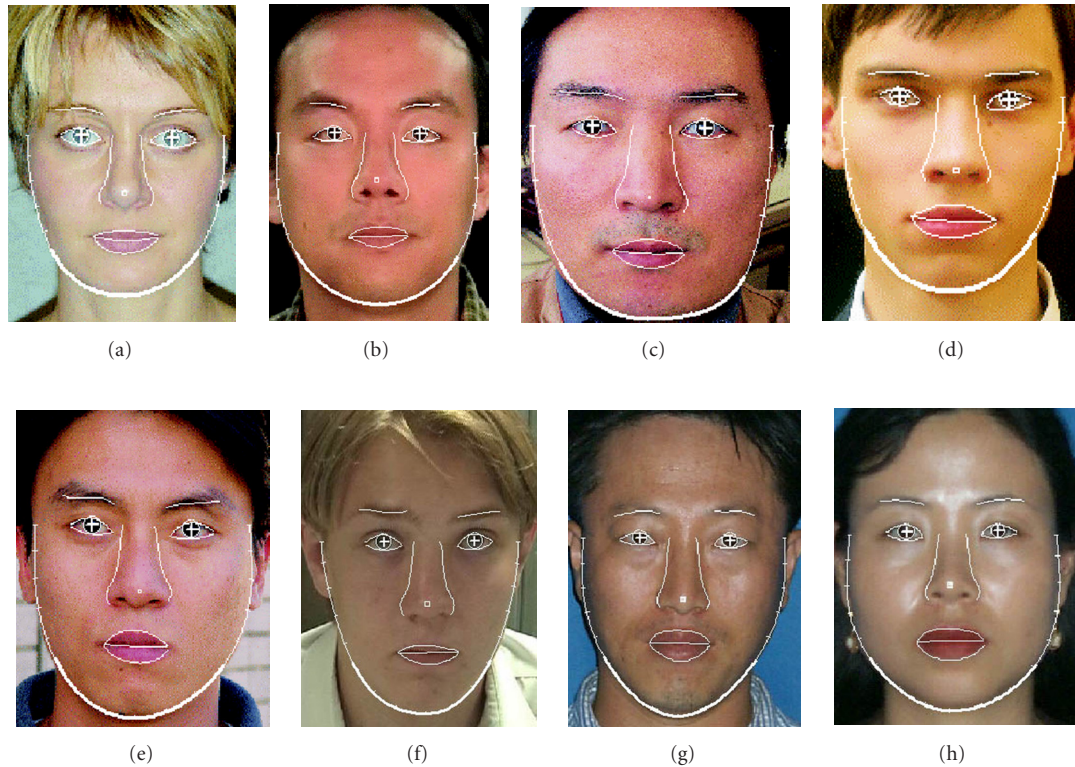
FIGURE 21: Final results of frontal feature detection.

developed on a Wintel platform using Microsoft Visual. Net and Visual C++ 6.0.

In order to evaluate the performance of the proposed modeling system, intensive experiments are carried out on many test samples. A total of 35 people of various genders and races are tested. In addition, the imaging condition is not calibrated. All the photos were taken under normal illumination conditions in a normal office. The intermediate results in feature detection are shown throughout the paper. In this section, the final results of the frontal feature extraction, profile feature extraction, and the final model generation are described.

The final detection results of the frontal and profile features are shown in Figures 21 and 22, respectively. It is shown that almost all the particular features are detected quite accurately and automatically, illustrating the robustness of the proposed algorithms. One exception is the ear inside the feature shown in Figure 14b, which is specified manually. Note that the profile features on the top and back head cannot be detected automatically due to the hair, and are manually adjusted by the user with a simple interaction. The 3D models created are presented in Figures 23, 24, and 25, where the test samples are grouped according to the race and gender. It is also shown that the system produces satisfactory results under a variety of genders, races, and imaging conditions. The generated models are quite convincing and have been proven subjectively that they are quite similar to the real test people. Note that the generic ear shape is used for several

models, because the shape deformation algorithm performs poorly for this part. An obvious defect in the created models is the uniform hair shape. Currently we are working on a hair-modeling tool to generate more realistic hair.

The computational complexity is quite tolerable. The whole model creation takes 2~3 minutes on a Pentium IV 2.5 GHz PC, including all the optional manual corrections.

7. CONCLUSION AND FUTURE WORK

This paper describes an image-based 3D face modeling system, which produces a convincing head model. The system takes the frontal and profile images as the input and generates a 3D head model by customizing the generic model. The system was developed to be used by an ordinary user with inexpensive equipment, and it was highly automatic and robust in different picture-taking conditions.

The key to the proposed system is the robust feature detection module, which is highly customized and optimized for face modeling purposes. The detected facial features were fed into the shape deformation module, in which the generic model was deformed to have facial features identical to the ones in the image.

Future research will include an extension of the current system to have facial animation functionality such as emotional expressions and talking. For example, expression cloning [33] or ratio images [34] can be used to upgrade a static head to an animated one. Moreover, a variety of
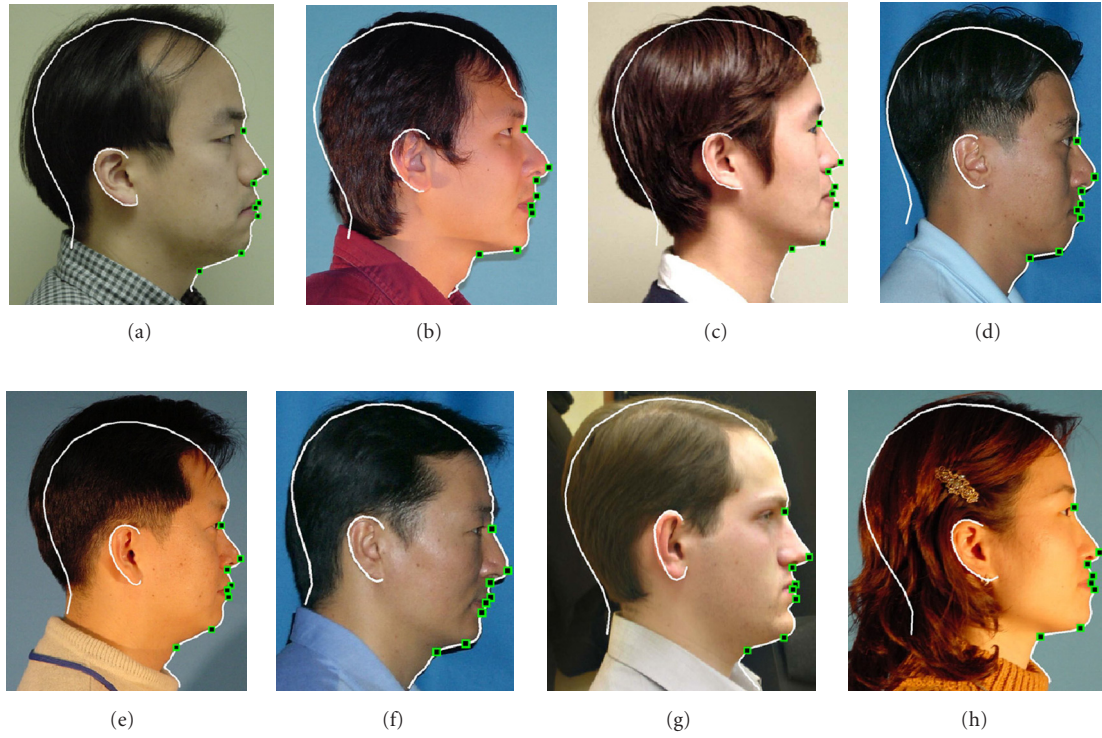
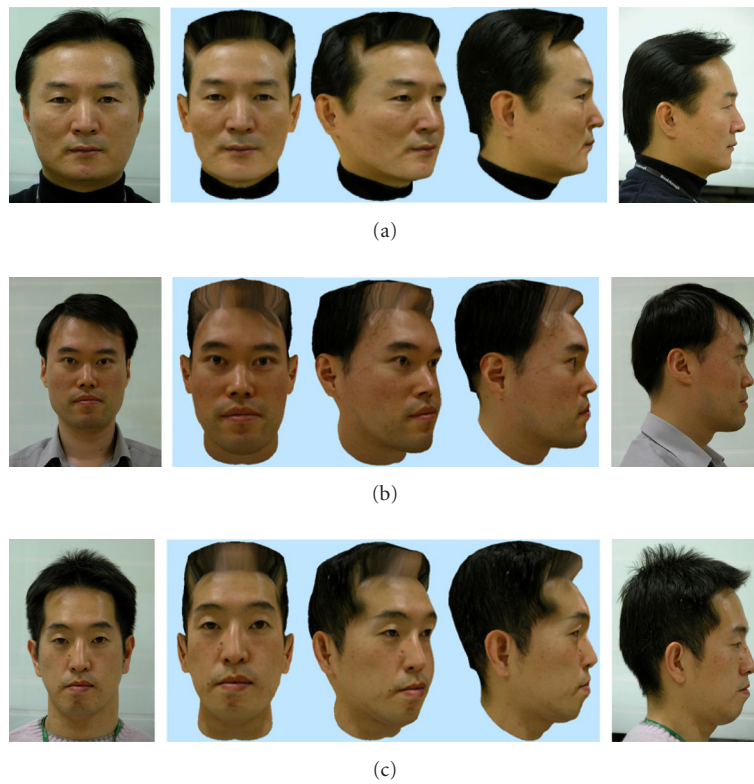FIGURE 22: The results of profile curves, profile fiducial points, and ear detection.

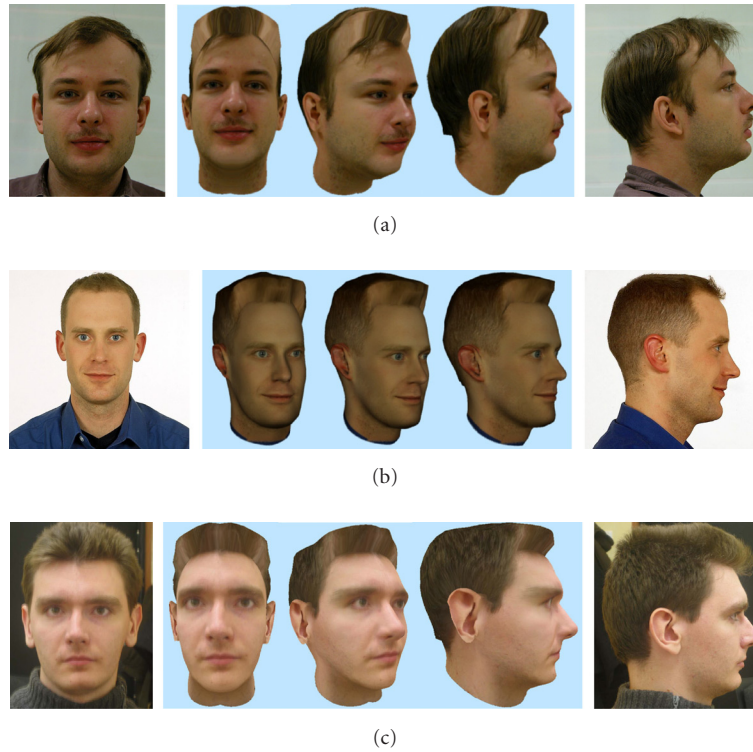FIGURE 23: Final face models (Oriental men) generated by the proposed system.

(a)

(b)

(c)

FIGURE 24: Final face models (Western men) generated by the proposed system.
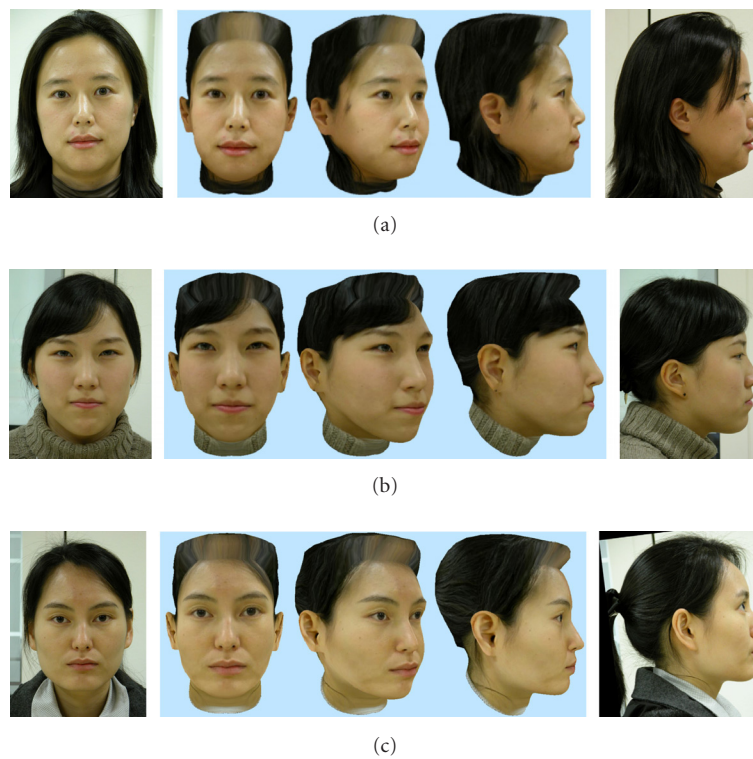

(a)

(b)

(c)

FIGURE 25: Final face models (Oriental women) generated by the proposed system.

possible applications in the field of computer games, virtual reality, video conference, and 3D character animation can benefit from the proposed face modeling system.

REFERENCES

[1] A. D. Linney, J. Campos, and R. Richards, "Non-contact anthropometry using projected laser line distortion: Three-dimensional graphic visualisation and applications," *Optics and Lasers in Engineering*, vol. 28, no. 2, pp. 137–155, 1997.

[2] Z. Chen, S.-Y. Ho, and D.-C. Tseng, "Polyhedral face reconstruction and modeling from a single image with structured light," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 864–872, 1993.

[3] D. Ivanov, V. Lempitsky, A. Selivanov, and Y. Kuzmin, "High-quality head model calibration pipeline," in *Proc. International Conference on Computer Graphics and Vision (GraphiCon '02)*, Nizhny Novgorod, Russia, September 2002.

[4] K. Lee, K. Wong, S. Or, and Y. Fung, "3-D face modeling from perspective-views and contour-based generic-model," *Real-Time Imaging*, vol. 7, no. 2, pp. 173–182, 2001.

[5] B. Dariush, S. B. Kang, and K. Waters, "Spatiotemporal analysis of face profiles: detection, segmentation, and registration," in *Proc. 3rd IEEE International Conference on Automatic Face and Gesture Recognition (FG '98)*, pp. 248–253, Nara, Japan, April 1998.

[6] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin, "Synthesizing realistic facial expressions from photographs," in *Proc. International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*, pp. 75–84, Orlando, Fla, USA, July 1998.

[7] Z. Liu, Z. Zhang, C. Jacobs, and M. Cohen, "Rapid modeling of animated faces from video," *Journal of Visualization and Computer Animation*, vol. 12, no. 4, pp. 227–240, 2001.

[8] Y. Shan, Z. Liu, and Z. Zhang, "Model-based bundle adjustment with application to face modeling," in *Proc. 8th IEEE International Conference on Computer Vision (ICCV '01)*, vol. 2, pp. 644–651, Vancouver, British Columbia, Canada, July 2001.

[9] R. Lengagne, P. Fua, and O. Monga, "3-D stereo reconstruction of human faces driven by differential constraints," *Image and Vision Computing*, vol. 18, no. 4, pp. 337–343, 2000.

[10] C. J. Kuo, R.-S. Huang, and T.-G. Lin, "3-D facial model estimation from single front-view facial image," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 3, pp. 183–192, 2002.

[11] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3-D faces," in *Proc. International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*, pp. 187–194, Los Angeles, Calif, USA, August 1999.

[12] S. Romdhani and T. Vetter, "Efficient, robust and accurate fitting of a 3D morphable model," in *Proc. 9th IEEE International Conference on Computer Vision (ICCV '03)*, pp. 59–66, Nice, France, October 2003.

[13] P. Fua, "Regularized bundle-adjustment to model heads from image sequences without calibration data," *International Journal of Computer Vision*, vol. 38, no. 2, pp. 153–171, 2000.

[14] R. Enciso, J. Li, D. Fidaleo, T. Kim, J. Y. Noh, and U. Neumann, "Synthesis of 3-D faces," in *Proc. 1st USF International Workshop on Digital and Computational Video (DCV '99)*, pp. 8–15, Tampa, Fla, USA, December 1999.

[15] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 1, pp. 34–58, 2002.

[16] B. D. Zarit, B. J. Super, and F. K. H. Quek, "Comparison of five color models in skin pixel classification," in *Proc. International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS '99)*, pp. 58–63, Kerkyra, Greece, September 1999.

[17] A. Yilmaz and M. Shah, "Automatic feature detection and pose recovery for faces," in *Proc. 5th Asian Conference on Computer Vision (ACCV '02)*, pp. 284–289, Melborne, Australia, January 2002.

[18] L. Jordao, M. Perrone, J. P. Costeira, and J. Santos-Victor, "Active face and feature tracking," in *Proc. International Conference on Image Analysis and Processing (ICIAP '99)*, pp. 572–576, Venice, Italy, September 1999.

[19] K. Toyama, "'Look, ma - no hands!' handsfree cursor control with real-time 3d face tracking," in *Proc. Workshop on Perceptual User Interfaces (PUI '98)*, pp. 49–54, San Francisco, Calif, USA, November 1998.

[20] S. Baskan, M. M. Bulut, and V. Atalay, "Projection based method for segmentation of human face and its evaluation," *Pattern Recognition Letters*, vol. 23, no. 14, pp. 1623–1629, 2002.

[21] X. Zhu, J. Fan, and A. K. Elmagarmid, "Towards facial feature extraction and verification for omni-face detection in video/images," in *Proc. IEEE International Conference on Image Processing (ICIP '02)*, vol. 2, pp. 113–116, Rochester, NY, USA, September 2002.

[22] Gu. C. Feng and P. C. Yuen, "Multi-cues eye detection on gray intensity image," *Pattern Recognition*, vol. 34, no. 5, pp. 1033–1046, 2001.

[23] J. Ahlberg, "A system for face localization and facial feature extraction," Tech. Rep. LiTH-ISY-R-2172, Linkoping University, Linkoping, Sweden, 1999.

[24] M. Kampmann and L. Zhang, "Estimation of eye, eyebrows and nose features in videophone sequences," in *Proc. International Workshop on Very Low Bitrate Video Coding (VLBV '98)*, pp. 101–104, Urbana, Ill, USA, October 1998.

[25] L. Yin and A. Basu, "Nose shape estimation and tracking for model-based coding," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP '01)*, vol. 3, pp. 1477–1480, Salt Lake City, Utah, USA, May 2001.

[26] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *International Journal of Computer Vision*, vol. 8, no. 2, pp. 99–111, 1992.

[27] M. Kampmann, "Estimation of the chin and cheek contours for precise face model adaptation," in *Proc. International Conference on Image Processing (ICIP '97)*, vol. 3, pp. 300–303, Santa Barbara, Calif, USA, October 1997.

[28] P. J. Burt, T. H. Hong, and A. Rosenfeld, "Segmentation and estimation of image region properties through cooperative hierarchical computation," *IEEE Trans. Syst., Man, Cybern.*, vol. 11, no. 12, pp. 802–809, 1981.

[29] R. Nevatia and K. R. Babu, "Linear feature extraction and description," *Computer Graphics and Image Processing*, vol. 13, pp. 257–269, 1980.

[30] J. C. Carr, R. K. Beatson, J. B. Cherrie, et al., "Reconstruction and representation of 3D objects with radial basis functions," in *Proc. International Conference on Computer Graphics and*

Interactive Techniques (SIGGRAPH '01), pp. 67–76, Los Angeles, Calif, USA, August 2001.

[31] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Comput. Graph. Appl.*, vol. 21, no. 5, pp. 34–41, 2001.

[32] P. J. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics," *ACM Trans. Graphics*, vol. 2, no. 4, pp. 217–236, 1983.

[33] J. Y. Noh and U. Neumann, "Expression cloning," in *Proc. International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pp. 277–288, Los Angeles, Calif, USA, August 2001.

[34] Z. Liu, Y. Shan, and Z. Zhang, "Expressive expression mappings with ratio images," in *Proc. International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pp. 271–276, Los Angeles, Calif, USA, August 2001.

In Kyu Park received the B.S., M.S., and Ph.D. degrees from Seoul National University (SNU), Seoul, Korea, in 1995, 1997, and 2001, respectively, all in electrical engineering and computer science. From September 2001 to March 2004, he was a member of technical staff in the Multimedia Laboratory, Samsung Advanced Institute of Technology (SAIT), where he was involved in several projects on computer graphics and multimedia applications. Also, he has been actively involved in MPEG-4 and MPEG-7 standardization activities. Since March 2004, he has been with the School of Information and Communication Engineering, Inha University, Korea, where he is currently a full-time Lecturer. Dr. Park's research interests include the joint area of 3D computer graphics, computer vision, and multimedia application, especially 3D shape reconstruction, range data processing, image-based modeling/rendering, and 3D face modeling. He is a Member of IEEE, IEEE Computer Society, and ACM.

Hui Zhang received the B.S. degree in applied mathematics and the Ph.D. degree in computer science, both from Tsinghua University, China, in 1997 and 2003, respectively. In February 2003, he joined the Samsung Advanced Institute of Technology, Yongin, Korea, as a researcher. Dr. Zhang's research interests are in the areas of computer vision, computer graphics, pattern recognition, and multimedia application, especially human face modeling, facial animation, and 3D object retrieval. He is a Member of IEEE.

Vladimir Vezhnevets received the "Diploma of Higher Education" with honors from Moscow State University, Russia, in 1999, and the Ph.D. degree in applied mathematics and computer science in 2004. In December 2002, he joined the Graphics and Media Lab, Computer Science Department, Moscow State University, as a researcher. His research interests are concentrated in the areas of computer vision and image processing; especially the problems of 3D shape reconstruction, face detection/tracking, image enhancement, and computer vision for human-computer interaction.