3D virtual simulator for breast plastic surgery

By Youngjun Kim*, Kunwoo Lee and Wontae Kim

We have proposed novel 3D virtual simulation software for breast plastic surgery. Our software comprises two processes: a 3D torso modeling and a virtual simulation of the surgery result. First, image-based modeling is performed in order to obtain a female subject’s 3D torso data. Our image-based modeling method utilizes a template model, and this is deformed according to the patient’s photographs. For the deformation, we applied procrustes analysis and radial basis functions (RBF). In order to enhance reality, the subject’s photographs are mapped onto a mesh. Second, from the modeled subject data, we simulate the subject’s virtual appearance after the plastic surgery by morphing the shape of the breasts. We solve the simulation problem by an example-based approach. The subject’s virtual shape is obtained from the relations between the pair sets of feature points from previous patients’ photographs obtained before and after the surgery. Copyright © 2008 John Wiley & Sons, Ltd.

Received: 25 June 2008; Accepted: 25 June 2008

KEY WORDS: 3D human modeling; image-based modeling; breast surgery; simulation

Introduction

Recently, plastic surgeries have created substantial interest. Among them, breast augmentation is one of the most commonly performed cosmetic surgical procedures. According to the American Society of Plastic Surgeons, breast plastic augmentation is the top surgical procedure, and the number of surgeries continues to increase in the United States.1 Since these breast plastic surgeries are actively performed, patients need a new type of simulation software to view the results prior to the surgery. Preoperative simulation software can be used as a communication tool, for (1) minimizing doctor–patient misunderstandings, (2) finding the optimal way of operation, (3) getting rid of the patient’s fear about the operation, (4) selecting the most desired figure, and so on. Surgeons can also persuade patients to undergo the operation via the simulating tool. Meanwhile, in computer graphics field many investigations on 3D human modeling and its applications have been undertaken. Various computer graphics technologies involving 3D human models have been introduced and employed in many other industries. Once these technologies are leveraged, we can successfully implement a 3D simulator for breast plastic surgeries. From this viewpoint, as well as the demand of a breast implant company and some surgeons, we have attempted to develop a realistic and intuitive 3D simulator for breast augmentation surgery. We introduce our 3D human modeling and virtual simulation method in the remaining part of this paper.

Related Work

• 2D breast surgery simulator
Many 2D simulation programs have already been developed.2–4 Image manipulation software provides tools that can be used to warp, stretch, shrink, and smooth features. They make cosmetic changes to digital images of the human face and body. Although such software can be used by many cosmetic clinics, they lack realism and are mostly unnatural. They are no more than image editing programs that collect suitable functions for simulation.

• 3D breast modeler and simulator
Seo et al.5 proposed a breast modeler based on the analysis of breast scans. Given a set of 28 scanned breasts, their breast modeler can be used to generate

*Correspondence to: Y. Kim, Human-Centered CAD Lab, Seoul National University, Shilim 9-Dong, Gwanak-Gu, Seoul, Korea. E-mail: yjkim@cad.snu.ac.kr

Copyright © 2008 John Wiley & Sons, Ltd.
the breast shape using principal component analysis (PCA). The subject’s shape is obtained by inputting the attribute parameters. It is an efficient way for modeling breasts. However, it needs abundant 3D scan data of nude female breasts to obtain a feasible result. Moreover, they did not consider photo-mapping to the reconstructed model.

Balaniuk et al.6 implemented a 3D breast simulator using the radial elements method (REM), which is a type of finite element method. They simulated the deformation and dynamic motion of a breast. They obtained good results, but they used a 3D scanner to get the subject’s body data. Further, they simulated only the one side of breast excluding other sections of the torso.

INAModel7 is a breast visualization tool available on the Internet. It interactively shows the 3D representation of the female body model, which can be manipulated to simulate the results of a breast augmentation procedure with various shapes and sizes. The biggest drawback of INAModel is that it provides simulation using a prepared model instead of the actual subject. The subject feels that the virtual model is different from her, and it detracts from reality.

### Overview

Our software has two main parts: a 3D modeler of a female torso and a virtual simulator of the person’s shape after breast surgery. Figure 1 shows the overview of our software.

Three-dimensional scanning is the optimal method to obtain the patient’s model data. However, a 3D scanner is prohibitively expensive for use in a general clinic. Therefore, we adopted image-based 3D modeling. Such a modeling method requires several orthogonal photographs of the subjects. Then, according to the input feature points on the photographs, a template 3D model is morphed by using our deforming methods. Detailed explanations of our modeling methods are described in the next section.

From the reconstructed patient’s model, virtual simulation of the post-surgery data is performed. We implemented this simulation using an example-based method. Feature points of the breast are moved to reconstruct the 3D model directly from orthogonal 2D images leveraging a 3D template data. So far, an image-based modeling method specific to the breasts has not been developed.

### Linear combination model

Blanz and Vetter18 introduced a morphable face model for reconstructing a new face and manipulating it according to changes in certain facial attributes. They modeled a new face by forming linear combinations of 200 scanned face models. Annotated facial attributes are used to define the shape and texture vectors, and they are added to or subtracted from the face. Hwang et al.19,20 also utilized the linear combination model. They applied it for reconstructing 2D face data from partially damaged images or from a small number of feature points. Based on their idea, we use statistical prediction based on exemplar cases.
new positions by learning from the exemplar database. Then, all the mesh vertices are mapped by interpolating the moved feature points. To obtain a statistically derived result, we utilized photographs of 30 patients before and after the plastic surgery. The following section presents our example-based simulation methods. The remaining sections discuss the results of our work.

**Image-Based 3D Torso Body Modeling**

**Template Model**

Generally, in image-based modeling, a 3D template model is commonly used in order to compensate for the lack of information from 2D images. The template model requires a typical and representative shape. We used a commercial modeling tool, Maya 7.0, to make the 3D template model. Thereafter, we deformed the template model according to the relations of the feature points of the template model and those from the images. In this paper, we denote the template feature points as $P_T$ and the calculated feature points from the images as $P_C$. We assigned the position of $P_T$ on the 3D template by manual selection. Several template models were prepared, and the most appropriate one is selected among them.

**Feature Point**

Feature points are necessary and important prerequisites for our image-based modeling method. We referred to Lee et al.'s definition, and excluded some points and defined some additional points for our purpose. The number of feature points is 10 for both left and right breasts. For the abdominal part, we defined 18 feature points: six points each for the left, right, and central waist curves. Additional points such as the navel, armpit, shoulder, and so on are also defined. Table 1 and Figure 2 show the definitions of feature points.

In our image-based 3D torso modeling, we use one front-view and two side-view photographs of the subject. We need both left- and right-view images in order to reconstruct unsymmetrical breasts. A back-view photograph is optionally added, if needed, for photo-mapping. We define the feature points from these orthogonal images. A white or blue background is recommended because we detect the body curves via skin detection.

After taking the photographs of the subject, we assign feature points on them. All these points or each point can be scaled and translated easily using our GUI. Some

<table>
<thead>
<tr>
<th>FeatPt</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>Upper breast point (UBP), same y-coord. as LAP on $C_{bf}$ &amp; $C_{bs}$</td>
</tr>
<tr>
<td>P1</td>
<td>Mid-point of $P0$ &amp; $P2$ on $C_{bs}$</td>
</tr>
<tr>
<td>P2</td>
<td>Bust point (BP) on $C_{bf}$ &amp; $C_{bs}$</td>
</tr>
<tr>
<td>P3</td>
<td>Mid-point of $P2$ &amp; $P4$ on $C_{bs}$</td>
</tr>
<tr>
<td>P4</td>
<td>Bottom breast point (BBP) on $C_{bf}$ &amp; $C_{bs}$</td>
</tr>
<tr>
<td>P5</td>
<td>Mid-point of $P0$ &amp; $P6$ on $C_{bs}$</td>
</tr>
<tr>
<td>P6</td>
<td>Outermost point on $C_{bf}$</td>
</tr>
<tr>
<td>P7</td>
<td>Mid-point of $P6$ &amp; $P4$ on $C_{bf}$</td>
</tr>
<tr>
<td>P8</td>
<td>Mid-point of $P4$ &amp; $P9$ on $C_{bf}$</td>
</tr>
<tr>
<td>P9</td>
<td>Inner breast point (IBP) on $C_{bf}$</td>
</tr>
<tr>
<td>FNP</td>
<td>Front neck point</td>
</tr>
<tr>
<td>SP</td>
<td>Shoulder point (SP (LSP, RSP)), same y-coord. as FNP</td>
</tr>
<tr>
<td>CSP</td>
<td>Center of shoulder point (CSP (LCSP, RCSP)) on $C_{bf}$, mid-point of FNP &amp; SP in x-dir.</td>
</tr>
<tr>
<td>OBP</td>
<td>Outer breast point (OBP (LOBP, ROBP)) on $C_{bf}$, mid-point of $P0$ &amp; $P4$ in y-dir.</td>
</tr>
<tr>
<td>WP1</td>
<td>Same y-coord. as $P4$, on $C_w$ or $C_c$</td>
</tr>
<tr>
<td>WP2-WP5</td>
<td>Equally dividing points between WP1 and WP6 in y-dir. on $C_w$ or $C_c$</td>
</tr>
<tr>
<td>WP6</td>
<td>Same y-coord. as NP, on $C_w$ or $C_c$</td>
</tr>
<tr>
<td>FCP</td>
<td>Front center point, mid-point of LBP9 &amp; RBP9</td>
</tr>
<tr>
<td>NP</td>
<td>Navel point, the same point as WP6 on $C_c$</td>
</tr>
</tbody>
</table>

Table 1. Definition of feature points. $l_f$, front-view image; $l_s$, side-view image; $C_{bf}$, breast curve in $l_f$; $C_{bs}$, breast curve in $l_s$; $C_w$, waist curve in $l_f$; $C_c$, belly curve in $l_s$.
Figure 2. Definition of feature points (R, right; L, left; C, center).

Points are constrained by the definitions listed in Table 1, and many of them are automatically positioned. For example, right shoulder point (RSP) and left shoulder point (LSP) have the same y-coordinates as front neck point (FNP). Their coordinates in the front-view image are set at the outermost positions of the body at the FNP’s y-level. Among the feature points, front center point (FCP) cannot be seen from the side view. Therefore, the z-position of this feature point is set on the side image by using the user’s estimate.

The feature points P0, P2, and P4 in the front-view image are assumed to lie on the same plane in the 3D coordinate system. These points are also defined in the side view. Internally, the side-view images are normalized according to the height of the breast, from P0 to P4, in the front-view image. All the feature points’ coordinates are calculated from the normalized orthogonal images. Their x-coordinates are obtained from the front image, and the z-coordinates are computed from the side image. The y-coordinates are common values in the front and side images.

WP1-WP6 are detected automatically using skin detection. We used YCbCr color space for the skin detection. We collate only the pixels that are within the skin color range of the Cb and Cr values. We only handle the Cb and Cr values because we are interested in the section invariant to the illumination intensity in order to facilitate skin detection. In this detection process, we have many small holes in the interior of the body. These holes act as obstacles in finding the body’s curves; therefore, we fill these holes if they are smaller than a threshold size proportional to the image size. Figure 3 shows the skin detection result and feature points input.

**Global Deformation**

In the global deformation step, we compute the affine transformation matrix to match the template model to
the feature points calculated from the images as closely as possible. The affine transformation includes translation, rotation, and scaling. To minimize the sum of the squared error between $P_C$ and $P_T$, we use the procrustes analysis. This method is computationally simple and stable. Global deformation is an auxiliary step, which yields good initial conditions for local deformation.

$$\min E(S, R, T) = \sum_{i=1}^{n} (p_{C,i} - p_{T,i})^2$$

(1)

where $n$ is the number of feature points for breast part, $p_{C,i}$ the $i$th feature point calculated from orthogonal images, $p_{T,i}$ the $i$th feature point of template model after global deformation, $P_T = (p_{T_1}, p_{T_2}, p_{T_3})^T = S \cdot R(p_{T_0}, p_{T_1}, p_{T_2})^T + T$, $P_{T_0}$ the template model’s feature points before global deformation, $S$ the scaling factor, $R$ the rotation matrix, and $T$ is the translation vector.

The procedures of the global deformation are as follows:

(a) Normalize $P_C$ and $P_{T_0}$.
(b) Let $A = P_C^T \cdot P_{T_0}$.
(c) Singular value decomposition of $A$.

$LDM = \text{SVD}(A)$

(d) Compute the rotation matrix $R = ML^T$.

(e) Compute the scaling factor.

$$S = \sum \text{diagonal}(D) \times \frac{\|P_C\|^2}{\|P_T\|^2}$$

(f) Compute the translating vector.

$$T = \text{mean}(P_C) - (S)\text{mean}(P_{T_0})(R)$$

(g) Transform the entire 3D template model’s vertices using $S$, $R$, and $T$.

An example global deformation result is shown in Figure 4.

**Figure 4.** Before and after global deformation (dot: $P_T$, cross: $P_C$).

### Local Deformation

After global rigid deformation roughly aligns the template model to $P_C$, local deformation refines and generates a more realistic model. For this process, an interpolation function is estimated, and the rest of the points are mapped using this function. Among the interpolation methods, radial basis functions (RBF) are a powerful technique for performing interpolation in a multidimensional space. The basis function, $R(d)$, depends only on the distance from the feature points, which are thus called radials. RBF construct the interpolants as a linear combination of the basis functions and then they can be used to determine the coefficients of the basis functions.

$$f(p) = \sum_{i=1}^{n} w(i)R(\|p - p(i)\|)$$

(2)

where $n$ is the number of feature points, $f(p)$ the transformed vertex through local deformation, $w(i)$ the coefficients of basis functions, $p(i)$ the model’s feature points, and $p$ is each vertex of generic model (input point).

The most common RBF, $R(d)$, are as follows:

- $R(d) = d$ (linear)
- $R(d) = d^3$ (cubic)
- $R(d) = d \log(d)$ (thin plate spline)
- $R(d) = (d^2 + c^2)^{\nu/2}$ (multiquadric)
- $R(d) = e^{-d^2/c^2}$ (Gaussian)

We investigated these RBF and selected the most appropriate one:

$$R(d) = e^{-d^2/50^2}$$

(3)

To find the coefficients of the basis functions $w(i)$, we use Equation (3) and let $f(P_T)$ equal to $P_C$.

$$p_{C,x}^j = f_x(p_{T,x}^j) = \sum_{i=1}^{n} w_x(i)R(\|p_{T,x}^j - p(i)\|)$$

(4)

where $p_{C,x}^j$ is the $x$-coordinate of $j$th calculated feature point and $p_{T,x}^j$ is the $x$-coordinate of $j$th template model’s feature point.

Because we have $n$ pairs of $p_{C,x}^j$ and $p_{T,x}^j$, we can solve Equation (4) and obtain the coefficients of the basis functions.

$$[w_x(1) w_x(2) \ldots w_x(n)]^T = H^{-1} [p_{C,x}^1 p_{C,x}^2 \ldots p_{C,x}^n]^T$$

(5)

where $H$ is a matrix that consists of RBF values.
By using the same steps mentioned above, we deform along the $y$- and $z$-directions, too. We apply the RBF interpolation to all the vertices of the template mesh data and finally obtain the morphed data according to the subject. Figure 5 shows an example of local deformation result.

**Photo-Mapping**

For realism, the photographs of the subject are mapped onto the deformed mesh data. Because we have already assigned feature points on the images, the texture coordinates of each vertex can be easily calculated. The main issue in texture mapping is how to blend three or four orthogonal images. We verified the normal direction of the facets to determine the blending ratio. Here, let us assume that we use the front-, left-, and right-view images for texture mapping. The front- and side-view images are orthogonal; therefore, the blending ratio is calculated by using the normal vector's $z$-component. At this time, the left-view image should be used only for the left part ($x > 0$), and *vice versa* for the right part. Equation (6) explains our texture image blending method in detail.

\[ \begin{align*}
  i) & \quad x > 0; (i.e., \text{left part}) \\
   & \quad \text{if } \vec{n}_x > 0, \alpha_f = \vec{n}_z, \alpha_l = 1 - \vec{n}_z, \\
   & \quad \text{otherwise, } \alpha_f = 1, \alpha_l = 0 \\
  ii) & \quad x < 0; (i.e., \text{right part}) \\
   & \quad \text{if } \vec{n}_x < 0, \alpha_f = \vec{n}_z, \alpha_r = 1 - \vec{n}_z, \\
   & \quad \text{otherwise, } \alpha_f = 1, \alpha_r = 0 
\end{align*} \]

(6)

where $\vec{n}_x, \vec{n}_z$ are the $x$- and $z$- components of the normal vector, respectively. $\alpha_f, \alpha_l, \alpha_r$ are the blending ratios of the front-, left- and right-view images used in texture mapping, respectively.

**Example-Based Simulation**

The ultimate goal of this study is to obtain a virtually simulated result of the breast augmentation surgery. In order to accomplish this, we implemented an example-based simulation algorithm. In this section, we describe a virtual breast surgery simulation method by learning actual surgical cases.

**Exemplar Data Preparation**

Our simulation algorithm utilizes the photographs of patients who have already undergone plastic surgeries for breast augmentation. With the support of plastic surgeons, we could accumulate the patient data. Photographs of three orthogonal views of each patient were taken before the surgery and about 2 weeks postsurgery. Further, 30 patients participated in this task. We prepared a database of the sets of feature point pairs from these data. Among the feature points that we defined in the previous sections, only the feature points of the breast part are needed in the morphing process for the simulation. We have computed the 3D coordinates of...
the feature points, which are manually positioned on the photos, as aforementioned. The feature point sets in the database are normalized with the height of each breast, from P0 to P4, respectively.

**Basic Assumption and Procedures**

In order to leverage the database of surgical cases, we have assumed that the subject’s virtually simulated result of the appearance would be similar to those who have similar breast figures. We can estimate the breast shape post-surgery on the assumption that it would follow the result of the patients in the database. If we suppose a subject who has exactly the same figure as one of the exemplar data, she would have the same result as the exemplar data’s surgical result. Each person’s individual taste is neglected.

Since we have reconstructed our 3D model using the feature points, we can easily morph the model by changing the coordinates of the feature points. This is why we prepared the exemplar database with photographs, and not with 3D scan data. Three-dimensional scan data of patients are much more difficult to gather than photographs. After we move the positions of the feature points to the virtually simulated positions, all of the mesh vertices in the breast segments are moved accordingly. After morphing using our example-based simulation, additional modification is possible with our detailed morphing tool. In the entire simulation process, morphing is performed by applying the RBF interpolation method to the template model with the newly changed feature points.

**Linear Combination of Exemplar Data**

The main idea of the simulation is that any subject can be expressed as a linear combination form of the exemplar data. Blanz and Vetter\(^{18}\) have presented a morphable model for face modeling, and many implemented the linear combination model for various applications.\(^3,19,27\) From those motives, we express a subject’s feature point set \(\tilde{P}\) as a linear combination form of the surgery exemplar data prior to surgery \(P_i\) in the database.

\[
\tilde{P} = \sum_{i=1}^{m} \alpha_i P_i, \quad \sum_{i=1}^{m} \alpha_i = 1
\tag{7}
\]

where \(m\) is the number of exemplar data in database, \(\tilde{P}\) the combination model’s feature point set before surgery, \(P_i\) the \(i\)th patient data’s feature point set before surgery, and \(\alpha_i\) is the \(i\)th weighting factor for linear combination.

The optimal values of the weighting factors \(\alpha^*\) are those which minimize the sum of the squared error between \(\tilde{P}\) and \(P_c\).

\[
\alpha^* = \arg \min_{\alpha} E(\alpha)
\tag{8}
\]

The error function is given as

\[
E(\alpha) = \sum_{j=1}^{n} \left( P_{C,j} - \sum_{i=1}^{m} \alpha_i p_{i,j} \right)^2
\tag{9}
\]

where \(n\) is the number of feature points in each set, \(m\) the number of exemplar data in database, \(P_{C,j}\) the \(j\)th calculated feature point from the subject’s orthogonal images, and \(p_{i,j}\) is the \(j\)th feature point of \(i\)th patient data before surgery.

**Example-Based Simulation**

In this section, we describe how we predict the virtually simulated data using the linear combination model. We denote the \(i\)th patient’s feature point set of before surgery as \(P_i\) and that after surgery as \(P'_i\). Now, let the subject’s combination model’s feature point set be expressed as follows:

\[
\tilde{P} = \{ \tilde{p}_j, j = 1, \ldots, n \}, \quad P' = \{ p'_j, j = 1, \ldots, n \}
\tag{10}
\]

where \(n\) is the number of feature points in each set, \(\tilde{P}\) the combination model’s feature point set before surgery, and \(P'\) the combination model’s feature point set after surgery.

Once we find \(\alpha^*_i\), we apply them to the displacement vectors between \(P_i\) and \(P'_i\). The weighted sum of the relative displacements is the virtually simulated position of the subject’s feature points.

For each feature point \(p_j (j = 1, \ldots, n),\)

\[
\tilde{p}_j = p_j + \sum_{i=1}^{m} \alpha^*_i (p'_{i,j} - p_{i,j})
\tag{11}
\]

where \(p_{i,j}\) is the \(j\)th feature point of \(i\)th patient data before surgery, \(p'_{i,j}\) the \(j\)th feature point of \(i\)th patient data after surgery.
Figure 6. Subject’s reconstructed model before surgery (left) and its virtually simulated model after surgery (right). Both are displayed in wireframe mode and texture mode.

data after surgery, \( \tilde{p}_j \), the \( j \)th feature point of combination model before surgery, and \( \tilde{p}'_j \) is the \( j \)th feature point of combination model after surgery.

Equation (11) implies that each patient’s relative displacement during the surgery influences the simulation process according to the weighting factor. In other words, a patient, whose shape is similar to the subject, would have a higher weighting factor, and her change during the surgical procedures would be more influential as compared to the others. In our algorithm, the feature point’s relative displacement is directly related to the change during surgery. Finally, after we move each feature point to \( \tilde{p}'_j \), we apply RBF interpolation to the mesh vertices with regard to the new positions of the feature points \( P' \). Figure 6 illustrates an example simulation result.

**Detailed Morphing by User Input**

In case the user wants to modify the results, our software provides detailed morphing tools. Figure 7 shows an example.

- **Local smoothing**

For using the local smoothing tool, the region of interest in the mesh is first selected by mouse operation. Ohtake et al.’s\(^{28}\) smoothing algorithm has been applied in this tool. It provides a fairing effect to the selected region.

- **Local extrusion/intrusion**

The local extrusion/intrusion tool is required if the user wants to extract or depress the selected region. It is performed through an interpolation function such as linear, squared, root squared function, etc., with the user’s input parameters and displacement amount. The selected region’s mesh vertices are moved through the given interpolation function.

- **Morphing using feature point**

If required, this tool is available to morph the mesh by modifying each feature point’s position. When the user selects a feature point and moves it to a new position, the mesh is morphed according to the new feature point set. Here too, the mesh data are obtained through RBF interpolation.

**Results**

Figure 8 shows the result of our image-based modeling as compared to the front-view photographs. As shown in this figure, both 3D geometry and texture information of each subject are effectively reconstructed.
for visualization purposes. We analyzed the results by comparing them with the subject’s actual 3D scan data. Table 2 lists these results. RapidForm\textsuperscript{29} was used to analyze distance errors. The errors are rather excessive because of the limitations of the image-based approach. By considering only the orthogonal views of the body might make it difficult to reconstruct the area between the breasts, since the images do not contribute to any information on the depth for this particular area. This part of the body can only be properly reconstructed if the template is very similar to the subject’s data. Moreover, different postures of the subjects increased the errors. However, our result is sufficiently realistic to meet our original goal, which involves the virtual simulation of breast plastic surgery.

We evaluated the linear combination model by comparing it with the actual model. A patient’s data before surgery were reconstructed using our image-based modeling, and the result yielded was compared with that obtained from the linear combination model. Data for the linear combination model were obtained from the exemplar database without considering the patient’s

![Figure 8. Examples of eight female subjects. The 1st row images are the frontal photographs of actual subjects. The 2nd row illustrates the snapshots of the reconstructed 3D data, and the 3rd row, the virtually simulated data after surgery.](image)

![Figure 9. Comparison of the reconstructed model with the linear combination model (a) distance error, (b) reconstructed model, and (c) linear combination model.](image)
Table 2. Distance errors between the scanned data and the reconstructed data of the subjects (unit, mm; SD, standard deviation)

<table>
<thead>
<tr>
<th>No.</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>−14.9</td>
<td>12.9</td>
<td>−0.10</td>
<td>5.77</td>
</tr>
<tr>
<td>2</td>
<td>−8.95</td>
<td>13.1</td>
<td>0.96</td>
<td>4.87</td>
</tr>
<tr>
<td>3</td>
<td>−16.3</td>
<td>12.2</td>
<td>0.02</td>
<td>4.75</td>
</tr>
<tr>
<td>4</td>
<td>−21.7</td>
<td>18.1</td>
<td>0.42</td>
<td>7.22</td>
</tr>
<tr>
<td>5</td>
<td>−14.1</td>
<td>13.8</td>
<td>0.28</td>
<td>6.40</td>
</tr>
<tr>
<td>6</td>
<td>−15.5</td>
<td>19.5</td>
<td>0.09</td>
<td>4.84</td>
</tr>
<tr>
<td>7</td>
<td>−18.3</td>
<td>19.7</td>
<td>0.67</td>
<td>5.91</td>
</tr>
<tr>
<td>8</td>
<td>−13.1</td>
<td>19.8</td>
<td>0.10</td>
<td>5.60</td>
</tr>
</tbody>
</table>

data. The two models from the exemplar data and its model obtained from the linear combination should be identical; this is because the simulated model of the subject’s $P'_i$ value is predicted directly from the combination model $\tilde{P}'$. As shown in Figure 9, the two models are very similar.

We also evaluated the simulation results by cross-validation using all the examples in the database. Consequently, each example has been excluded and thereafter the simulation algorithm was tested. The results are shown in Figure 10. The results can be improved as we have more exemplar cases.

**Discussion**

In this paper, we propose an image-based female torso modeling and an example-based simulation method. First, some template models were prepared and the feature points were defined. We implemented several geometric deforming algorithms and a texture mapping method. Our image-based 3D modeling method enables users to get a subject’s 3D model without employing expensive hardware such as 3D scanners. Then, the subject’s appearance after breast surgery is simulated through an example-based algorithm. It is based on a database involving pair sets of feature points calculated from photographs obtained before and after surgery. We have developed this database from the photographs of actual patients with the help of surgeons. By using the relationship of the feature points in the photographs before and after surgery, our algorithm predicts the surgery results. We designed a linear combination model to reasonably apply the relationship derived from the database. If needed, a user can interactively refine the result by using the morphing tool. It takes about 1–2 minutes to obtain the simulated result from taking the photographs of the subject.

Our software enables surgeons to counsel patients with the simulated results before the cosmetic breast surgery. As preoperative simulation software, it has many benefits: minimizing patient–doctor misunderstandings, finding the optimal way of operation, getting rid of the patient’s fear about the operation, selecting the optimal figure, and so on.

With regard to future work, an automatic way of positioning the feature points needs to be developed. Currently, only the feature points related to the profiles of the body are automatically detected. Further, it is necessary to enrich our database of procedure cases. We need to gather more exemplar data according to implant volumes as well as various implant types. Breast implants differ by shape, profile, and size. If we have sufficient amount of exemplar data in every category of implants, we would be able to simulate with regard to type, profile, and size of the implant. Thus far, this is not possible due to the lack of number of surgical cases in the database. As many surgeons use our software for counseling, we can accumulate more exemplar data, resulting in our software to become more robust.

![Figure 10. Cross-validation of simulation algorithm. Each example has been excluded and thereafter the simulation was tested. Fifteen out of total 30 patients’ data are shown here. First row: reconstructed model from before surgery’s photographs, 2nd row: virtually simulated model, 3rd row: reconstructed model from after surgery’s photographs.](image-url)
ACKNOWLEDGEMENTS

We thank Dr DongRak Lee and Dr SungJun Back for their operation data and advice. This work is financially supported by the Ministry of Education and Human Resources Development (MOE), the Ministry of Commerce, Industry and Energy (MOCIE), and the Ministry of Labor (MOLAB) through the fostering project of the Lab of Excellency. This research was also supported in part by SNU-IAMD.

References


Authors’ biographies:

Youngjun Kim is a PhD student in the School of Mechanical and Aerospace Engineering at Seoul National University in Korea. He received his BS(2001) and MS(2003) from Seoul National University. He also has been working at K&I Technology. His research interests include geometric modeling, computer vision, 3D scanning, volume data processing, and medical engineering.
Kunwoo Lee is a professor in the School of Mechanical and Aerospace Engineering at Seoul National University in Korea. He received his BS (1978) from Seoul National University, and MS (1981) and PhD (1984) in mechanical engineering from MIT. He was previously an assistant professor in the Department of Mechanical & Industrial Engineering, University of Illinois at Urbana-Champaign, a visiting associate professor in the Department of Mechanical Engineering at MIT, and a visiting professor in the Department of Mechanical Engineering at Stanford University. Currently, he is a Co-Editor in chief of the journal “Computer-Aided Design” and the president of Advanced Institute of Convergence Technology.

Wontae Kim received his BS (2004) and MS (2006) in the Department of Computer Science at Sogang University in Korea. His research interests include geometric modeling, volume rendering, GPGPU, and real time rendering. Currently, he is developing applications for volume graphics and 3D scanning system at K&I Technology.