Image-based modeling of 3D objects with curved surfaces

By Man Hee Lee and In Kyu Park*

This paper addresses an image-based method for modeling 3D objects with curved surfaces based on the non-uniform rational B-splines (NURBS) representation. The user fits the feature curves on a few calibrated images with 2D NURBS curves using the interactive user interface. Then, 3D NURBS curves are constructed by stereo reconstruction of the corresponding feature curves. Using these as building blocks, NURBS surfaces are reconstructed by the known surface building methods including bilinear surfaces, ruled surfaces, generalized cylinders, and surfaces of revolution. In addition to them, we also employ various advanced techniques, including skinned surfaces, swept surfaces, and boundary patches. Based on these surface modeling techniques, it is possible to build various types of 3D shape models with textured curved surfaces without much effort.

Introduction

Recent progress in multimedia technology has been focused on the creation, manipulation, and display of 3D graphics content, rather than conventional media such as images, videos, and sounds. Therefore, there is a necessity for advanced technologies which create and manage 3D graphics content more efficiently at reasonable cost in the areas of academia, industry, and consumer electronics.

The life-cycle of any multimedia content including 3D graphics content generally consists of four steps—authoring, storage, transmission, and rendering. There have been a lot of previous works involving each of these pipeline components. In order to efficiently store and transmit 3D graphic models, the techniques of data compression and streaming over the network have been actively developed as a natural extension to the corresponding techniques for image and video data. The 3D graphics community has developed a lot of excellent algorithms for real time rendering and global illumination, using hardware accelerators as the workhorses.

Among the four ingredients in the life-cycle pipeline, the authoring step remains the most underdeveloped area. Unlike images and videos, which people can capture using the common camera and camcorder, creating 3D graphics content still remains an area for specialized people such as 3D animators and graphics designers. Usually, professional designers create 3D graphics content using expensive commercial software, in what amounts to a time-consuming task requiring a great deal of effort, even for them. In this context, it is necessary to develop efficient techniques which allow graphics content to be created more easily, so that regular users can create and utilize their own content.

In order to overcome this problem, the image-based approach known as image-based modeling (IBM) has attracted a great deal of interest as an alternative method of creating 3D content. Image-based geometry modeling is based on 3D computer vision technology, in which only a few images are used as the input and the 3D geometry is reconstructed by stereo reconstruction methods. IBM allows for the fast acquisition of 3D shapes with a photorealistic texture without an exorbitant computational cost.

*Correspondence to: I. K. Park, School of Information and Communication Engineering, Inha University, 253 Yonghyun-dong, Nam-gu, Incheon 402-751, Korea. E-mail: pik@inha.ac.kr

Copyright © 2007 John Wiley & Sons, Ltd.
However, the traditional IBM algorithms mostly reconstruct 3D point geometry and its subsequent triangulation in order to construct the 3D mesh model. This approach has few problems, since it cannot capture precise curved geometry. That is, it produces a noisy triangulation of the 3D surface even though the surface is a mathematically continuous one such as a sphere, cylinder, or other type of surface. It would be much more precise and simple if the 3D surface were reconstructed and defined in mathematical form directly. Note that the traditional approach is more appropriate when modeling polyhedral shapes rather than curved shapes.

In this paper, we propose novel techniques for modeling the shape and texture of objects with curved shapes, using an image-based approach and efficient user interaction. The proposed approach is based on non-uniform rational B-splines (NURBS) surface reconstruction and texture mapping. Note that the NURBS surface can represent a wide variety of curved surfaces in exact mathematical form and its shape modification is highly interactive, which makes it possible to use it for the geometric primitive employed for image-based surface reconstruction.

The proposed surface modeling techniques include building methods for surfaces such as bilinear surfaces, ruled surfaces, generalized cylinders, and surfaces of revolution. In addition, we propose various advanced surface modeling techniques, including skinned surfaces, swept surfaces, and boundary patches. By reconstructing the materials, such as boundary curves and spines, in an image-based way and employing one of the NURBS surface building algorithms, it is possible to reconstruct the curved surface defined in the 3D NURBS surface. In addition, a view-dependent texture modeling technique is proposed to capture the texture of the NURBS surface consistently. Together with a few reconstructed and textured NURBS surfaces, the process of modeling the object is finalized to produce photorealistic 3D objects with arbitrarily curved surfaces. Note that the proposed technique can model polyhedral shapes too.

This paper is organized as follows. In the following two sections, previous works on IBM and the overall structure of the proposed approach are introduced. Subsequently, the proposed image-based NURBS surface modeling technique is described in detail. Then the experimental results are shown and discussed. In the last section, we give our conclusive remarks.

---

1 In this paper, ‘skinned’ has a different meaning from the same term in computer animation. This is one of the surface building methods in NURBS5.
provides a few 3D geometric primitives such as the plane, cube, cylinder, sphere, and disk; and deforms them to reconstruct the 3D shape. In order to represent curved surfaces, it provides a triangulation tool using a 3D points cloud which, although it is not perfect, can be used to represent a curve very precisely in a mathematical way. Canoma has an interesting feature that reconstructs the 3D structure from a hand-sketched or printed sheet of objects, by placing and deforming the predefined 3D primitives manually onto them.

However, since these techniques reconstruct the 3D shape using simple and predefined 3D geometric primitives, it is difficult to precisely reconstruct the shape of objects with a few curved surfaces which have an exact mathematical representation. The reconstructed shape usually suffers from noise if it is reconstructed in dense mesh form, whereas the noise could be removed if the surface were reconstructed in mathematical form directly. Therefore, in this paper, we propose an efficient approach to reconstruct 3D objects with a curved surface based on the NURBS surface, which can represent a variety of shaped surfaces in clear mathematical form.

Overview of the Proposed Image-Based Modeling System

The IBM technique proposed in this paper performs the reconstruction of 3D shapes with curved surfaces using a few calibrated images as the input. Once the corresponding curves in the input images are specified by the simple user interface, a 3D curve is reconstructed by employing the stereo reconstruction technique. The curves and surfaces used in this paper are modeled in a mathematical way using the NURBS representation, which enables us to build arbitrary curves with efficient interactive modification of their shape. Note that the data points to interpolate are easily inserted, removed, and moved by means of a simple user interface.

The proposed surface modeling algorithms are based on NURBS curve and surface algorithms. They utilize the basic and advanced surface modeling techniques, which will be addressed in the ‘NURBS Surface Modeling’ subsection. As a special type of surface, a polyhedron can also be reconstructed using 1st degree NURBS. Furthermore, disks as well as concave planar objects, which are common shapes in the real-world environment, can also be reconstructed. Photorealistic textures are captured from the input image using view-dependent acquisition and mapped onto the reconstructed surface.

The input images and reconstructed 3D curved surfaces, as well as the intermediate results, are managed in a project workspace, which can export and reuse the 3D model in a standard 3D graphic format, i.e., virtual reality modeling language (VRML).

Image-Based Reconstruction of 3D Curved Surfaces Using the NURBS

In this section, we describe the proposed curve and surface reconstruction techniques in detail.

Camera Calibration

In order to reconstruct 3D geometry, the camera parameters of each input image should be estimated beforehand. In our approach, it is assumed that the exchangeable image file format (EXIF) information is available for the input image. Note that most of recent digital cameras provide the EXIF information, which is embedded in the image file. For camera calibration, the internal camera parameters are fetched from the EXIF information. The external parameters are computed by employing the efficient 5-point algorithm.

Modeling and Representation of 2D Curves Using the NURBS

As a prerequisite to reconstructing a 3D curved surface, 3D curves need to be build in which each of them is reconstructed from a pair of corresponding 2D curves. In this paper, we use the NURBS curve to represent arbitrary curves. The NURBS curve is defined by the control points, $P_i$, the weight, $w_i$ of $P_i$, and the $p$th degree basis functions, $N_{i,p}(u)$, as follows.

$$ C(u) = \sum_{i=0}^{n} N_{i,p}(u) w_i P_i, \quad 0 \leq u \leq 1 $$
In which the basis function, $N_{i,p}(u)$, is defined over a knot vector, $U$ (2) where $m = p + n + 1$.

$$U = \{0, \ldots, 0, u_{p+1}, \ldots, u_{m-p-1}, 1, \ldots, 1\}$$ (2)

In our approach, in order to approximate the curved feature of the object in the input images, the data points are incrementally added and a NURBS curve is instantly computed to interpolate them. In this procedure, local bicubic interpolation is performed to create the NURBS curve. In Figure 1, it is shown that nine data points are inserted manually and the NURBS curves are reconstructed to approximate the contour of the clock’s side. This procedure is performed on each of the corresponding curves, which commonly exist in two or more input images.

**Reconstruction of 3D Curves Using Multiple Input Images**

In stereo reconstruction, the epipolar line simplifies the process of finding the pixel correspondence in different images, since it reduces the search area confined to the line. In our approach, it is assumed that the camera is pre-calibrated. Therefore, it is natural to use the epipolar constraint in finding the data points of the corresponding curve. Figure 2 shows the proposed procedure, in which the yellow line denotes the epipolar line of a data point projected onto the other input image. Since the clock’s side contour is being approximated, the next data point that should be specified would be the intersection of the epipolar line and the visual edge. By repeating this procedure, it is possible to obtain the correspondence set of the data points.

From these data points, we use stereo reconstruction to reconstruct the 3D data points. The reconstructed 3D data points are used to recover the 3D NURBS curve, which is the 3D reconstruction of the approximated 2D contour, by employing local bicubic interpolation.

**NURBS Surface Representation**

The NURBS surface with degrees $p$ and $q$ in the $u$ and $v$ directions, respectively, is defined by Equation (3).

$$S(u, v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}P_{i,j}}{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}}$$ (3)

In Equation (3), $P_{i,j}$’s denote the points on the control net and $w_{i,j}$ is the weight of $P_{i,j}$. $N_{i,p}(u)$ and $N_{j,q}(v)$ refer to
In this paper, the curved surfaces are reconstructed by choosing an appropriate NURBS surface modeling method, according to the surface characteristics. Based on the surface type observed from the images, a set of proper 3D geometric primitives is determined and reconstructed, from which a selected surface modeling is performed to yield the output surface.

Table 1 shows the various NURBS surface modeling methods which are employed in this paper. The bilinear surface is the NURBS surface reconstructed by bilinear interpolation of four corner points. In Figure 3, the reconstruction process of the bilinear surface is shown. The ruled surface is a linearly interpolated surface defined over two opposite boundary curves. Figure 4 shows the reconstruction process of the ruled surface.

Furthermore, the basis functions defined over the knot vectors, $U$ (4) and $V$ (5), where $r = p + n + 1$ and $s = q + m + 1$.

$$U = \{0, \ldots, 0, u_{p+1}, \ldots, u_{r-p-1}, 1, \ldots, 1\}_{p+1}$$

$$V = \{0, \ldots, 0, v_{q+1}, \ldots, v_{s-q-1}, 1, \ldots, 1\}_{q+1}$$

### NURBS Surface Modeling

In this paper, the curved surfaces are reconstructed by choosing an appropriate NURBS surface modeling method, according to the surface characteristics. Based on the surface type observed from the images, a set of proper 3D geometric primitives is determined and reconstructed, from which a selected surface modeling is performed to yield the output surface.

Table 1 shows the various NURBS surface modeling methods which are employed in this paper. The bilinear surface is the NURBS surface reconstructed by bilinear interpolation of four corner points. In Figure 3, the reconstruction process of the bilinear surface is shown. The ruled surface is a linearly interpolated surface defined over two opposite boundary curves. Figure 4 shows the reconstruction process of the ruled surface.

The mathematical details are described fully in Reference [5].
The generalized cylinder is the surface which is obtained by sweeping the section curve along the axis. Figure 5 shows the reconstruction process of the generalized cylinder. The surface of revolution is the surface which is obtained by revolving the generatrix curve around the axis. Figure 6 shows the reconstruction process of the surface of revolution. The skinned surface is an interpolating surface, which interpolates a set of section curves. Figure 7 shows the reconstruction process of the skinned surface. The swept surface is constructed by sweeping the section curve along the trajectory curve. Figure 8 shows the reconstruction process of the swept surface. Finally, the boundary patch is a surface which interpolates four boundary curves. Figure 9 shows the reconstruction process of the boundary patch.

In addition to the surface modeling techniques described above, three additional techniques are proposed in our approach, i.e., a perfect disk, a concave planar face, and a polygonal face. The perfect disk can be reconstructed from a perfect circle with proper tessellation into triangles, as shown in Figure 10.

In order to represent a concave planar face, which is in fact the generalized shape of a disk, we proposed a simple but effective method based on contour partition.
Figure 6. The reconstruction process of a surface of revolution. (a) Generatrix curve and axis, (b) reconstructed surface of revolution.

Figure 7. The reconstruction process of the skinned surface. (a) A set of section curves to interpolate, (b) reconstructed skinned surface.

Figure 8. The reconstruction process of a swept surface. (a) Section and the trajectory curves, (b) reconstructed swept surface.
and the ruled surface. The boundary of the concave face is partitioned manually into a couple of separate curves, followed by ruled surface modeling using the separate curves as input. Figure 11 shows an example of this procedure. It is observed that there is no seam at all between the clock’s side modeled by the proposed concave face and the clock’s front and back surface which is modeled by the generalized cylinder. This is due to the fact that they share a common 3D curve as the input.

Since a scene usually contains a mixture of objects with curved and planar surfaces, it is necessary to represent the polygonal face in the same framework as that used for modeling a curved surface. Based on the fact that a piecewise polyline can be represented by a NURBS curve of degree 1, it is possible to model a polygonal face or
Figure 11. The reconstruction process of a concave planar face. (a) The concave planar curve, (b) reconstructed concave planar face, and (c) the concave planar face joining continuously with a generalized cylinder.

In order to reconstruct the photorealistic model, the texture as well as the geometry should be captured in a 3D polyline, as well as to use them as the input for the NURBS surface modeling techniques. In Figure 12, a polyline is used as the section curve for the generalized cylinder and then the surface, which resembles a folding screen, is reconstructed as a result.

Figure 12. The reconstruction process of a polygonal face. (a) A polyline, (b) reconstructed polygonal face.

**View-Dependent Texture Modeling of the NURBS Surface**

In order to reconstruct the photorealistic model, the texture as well as the geometry should be captured in a...
Figure 13. Scene 1 for modeling a few individual objects with different surface types. The total procedure takes 5 minutes for an intermediately-skilled user. (a) 5 input images, (b) reconstructed 3D NURBS curves, (c) reconstructed 3D NURBS surfaces, and (d) reconstructed 3D models with texture mapping.
Figure 14. Scene 2 for modeling a single object with different surface parts. The total procedure takes 20 minutes for an intermediately-skilled user. (a) 6 input images, (b) reconstructed 3D NURBS curves, (c) reconstructed 3D NURBS surfaces, and (d) reconstructed 3D models with texture mapping.
consistent way. Therefore, for each reconstructed NURBS surface, photorealistic texture maps should be obtained from the input images. Since the NURBS surface has the inherent $u$-$v$ parametrization, it is natural to use these parameters as the texture coordinates. In our approach, each NURBS surface is reprojected onto the individual input images and the texel colors of the corresponding texture map are read from the pixel at the reprojected position.

In this procedure, it is common for each texel to be reprojected onto more than one image and it is necessary to determine the texel color among the multiple corresponding pixels. In our approach, a view-dependent texture modeling technique is employed to blend the pixel colors. Let $V_i$ be the viewing vector of the input image $i$ and $N_{u,v}$ be the normal vector of the NURBS surface with the parameters $u$ and $v$. The weight factor $r_i$, which accounts for the confidence measure of the texture confidence, is defined as

$$ r_i = V_i \cdot N_{u,v} \quad (6) $$

The texel color $TC_{u,v}$ at $(u, v)$ is computed by determining the weighted average of the pixel colors $C_i$'s, as defined in Equation (7)

$$ TC_{u,v} = \frac{\sum s_i C_i r_i}{\sum s_i r_i}, \quad s_i = \begin{cases} 1 & \text{if the image } i \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \quad (7) $$

In some cases, there is occlusion between the objects in the scene. Therefore, in this case, it is preferable to avoid using the image of the occluded object. In order to account for this, in Equation (7), the binary variable $s_i$ is included, which is zero if the image $i$ is excluded by the user, due to occlusion or other reasons.

**User-Interface and System Functionality**

All the ingredient modules are integrated in a project workspace, which includes the windows and graphical user interfaces for 2D images and the reconstructed 3D scene. The user-interface for the 2D image is designed to edit the NURBS curve easily by adding, moving, and deleting the data points using computer mouse. The user-interface for the 3D scene allows the standard rotation, translation, and zooming scheme. Overall, the whole system takes the set of input images and produces an output VRML file which contains the mesh geometry of the reconstructed 3D scene plus textures.

**Experimental Results**

In order to evaluate the performance of the proposed algorithm, the experiments are carried out on few real test senses. The experiments were performed on a 64-bit AMD Athlon 2.0 GHz CPU with 2 Gbyte main memory and an NVIDIA GeForce 6800 GPU.

**3D Objet Modeling**

In our experiments, three real scenes are tested. The curved objects in each test scene are modeled using the proposed image-based NURBS surface modeling techniques.

In Figure 13, a few individual objects with different shapes are modeled using five input images. The clock is reconstructed with a generalized cylinder and a concave planar face. The Coke can is reconstructed with a generalized cylinder and a disk. The bottle of milk is reconstructed with a surface of revolution and a disk. The bent paper is reconstructed with a skinned surface. The Chinese tea cup is reconstructed with a surface of revolution. Figure 13a shows the input images, and Figure 13b and c shows the reconstructed NURBS curves and surfaces, as the intermediate result. And, Figure 13d shows the final NURBS model with texture. The modeling time is about 5 minutes for an intermediately-skilled user.

In Figure 14, a complex object, i.e., a projector is modeled, which consists of different shaped surface parts. Modeling is performed with 11 ruled surfaces, a generalized cylinder, and a disk. Figure 14b and c shows the reconstructed NURBS curves and surface, and Figure 14d shows the final NURBS model with texture. The modeling time is about 20 minutes.

In order to show the reconstructed result of polyhedral shape, the bottle of milk is reconstructed with a generalized cylinder in which the section curve is a polyline. Figure 15a shows four input images. Figure 15b shows the reconstructed NURBS curves and surfaces, and Figure 15c shows the final NURBS model with texture. The modeling time is about 2 minutes.

The reconstructed 3D surface models can be exported in common graphics formats such as VRML. Figure 16 shows the exported 3D models from the reconstruction shown in Figure 13d, which is rendered in a common VRML viewer.

In Figure 17, a set of the reconstructed textures of the test models is shown. Figure 18 shows the texture mapping results using different texture acquisition methods. Figure 18a shows the result when the proposed
Figure 15. Scene 3 for modeling a polyhedral object. The total procedure takes 2 minutes for an intermediately-skilled user. (a) 4 input images, (b) reconstructed 3D NURBS curves and surfaces, and (c) reconstructed 3D models with texture mapping.

A view-dependent texture modeling technique is used, while Figure 18b shows the result when the nearest texture is used. It is observed that the texture mapping based on the proposed technique is better in terms of smooth color transition and consistency with the input images.

Quantitative Analysis

Although the reconstructed 3D models are observed to be photorealistic and consistent visually with the input images both in shape and texture, cross-checking is performed to measure the performance quantitatively.
Figure 16. The rendering result of the reconstructed objects in the VRML viewer.

Figure 17. The reconstructed textures.
In this experiment, the test scene shown in Figure 13 is used. First, the reconstructed 3D models are reprojected onto the input images to evaluate the amount of object region overlap. It is observed that the non-overlapping region is 6.79% of the area of the whole object region. Next, the difference of the color between the corresponding pixels is computed in PSNR. In Table 2, PSNR scores are shown for a few typical resolutions of texture. It is observed that it is appropriate to select the texture resolution similar to the resolution of the effective segmented region of the object in the input image.

Once there is occlusion between the objects in the input image, such an undesirable image can be excluded from the texture modeling by user interaction. Therefore, it is possible to avoid generating visually inconsistent texture of the occluded object. Table 3 shows the difference of PSNR when the texture is obtained with or without this option.

<table>
<thead>
<tr>
<th>Texture resolution</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>256 × 256</td>
<td>19.14</td>
</tr>
<tr>
<td>512 × 512</td>
<td>19.19</td>
</tr>
<tr>
<td>1024 × 1024</td>
<td>18.44</td>
</tr>
</tbody>
</table>

Table 2. Comparison of PSNR, different texture resolution

<table>
<thead>
<tr>
<th>Image selection method</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>All images</td>
<td>19.19</td>
</tr>
<tr>
<td>Selected images</td>
<td>18.28</td>
</tr>
</tbody>
</table>

Table 3. Comparison of PSNR, different image select method. The resolution is 512 × 512

Note that the quantitative error does not seem to be negligible at first glance. However, in most cases, the whole texture on a single NURBS patch is slightly shifted on the reprojected image. Therefore, the qualitative visual quality remains almost same as the original one.

Limitations

The performance of the proposed approach is dependent on the selection of the particular camera calibration algorithm. A small error in camera calibration would result in visible mismatch between the reprojected position and the original position in the input image. This is exemplified in Figure 14d, in which a small glitch is observed at the right side of the projector model. This is due to the different calibration error for the images used in modeling the front and the side part of the projector. Note that the 5-point algorithm\(^\text{17}\) is used in our approach, since there is no need to use the calibration checker board. However, it is obvious that the more accurate is the calibration method used (for
example Zhang’s algorithm\textsuperscript{18}, the more accurate will be the surface modeling result.

Another limitation of the proposed approach is that it depends on the surface texture of the object when fitting feature curve on the input image. It means that it is not always possible to find the visible feature curve if there is no apparent texture on the object surface. Because of this, the revolving curve is painted manually on the surface of the milk container, as shown in Figure 13a. The problem can be solved properly if dense stereo matching is applied on the textures within the boundary curves and the result is fit with the NURBS surface. Note that it is considered for future research.

Conclusions and Future Works

In this paper, we proposed an image-based method for reconstructing 3D shapes with curved surfaces using the NURBS representation. Starting from a few calibrated images, the user specifies the corresponding curves by means of an interactive user interface. Then, the 3D curves are reconstructed by employing stereo reconstruction of the control points followed by the local interpolation of the NURBS curve. The proposed surface modeling techniques include surface building methods such as bilinear surfaces, ruled surfaces, generalized cylinders, and surfaces of revolution. In addition, we also propose various advanced surface modeling techniques including skinned surfaces, swept surfaces, and boundary patches. Based on these surface modeling techniques, it is possible to build various types of 3D shape models with curved surfaces without much effort. The constructed 3D shape model with curves and curved surfaces can be exported in VRML format, making possible for it to be used in different 3D graphics software programs. The proposed techniques can be used in general 3D graphics and CAD authoring tools, as well as in authoring tools used for augmented reality.

Although the proposed techniques provide an efficient method of modeling 3D objects with curved surfaces, they still rely on user interaction. That is, the corresponding curves have to be specified manually. The future direction of our research will be to focus on the automation of this part of the process. By developing curve matching algorithms which can deal with partial curves, the correspondence specification can be done automatically. In addition to this, the semantic information of the objects to be modeled can be used efficiently, yielding curve matching algorithms based on the active shape model (ASM)\textsuperscript{19} and active appearance model (AAM)\textsuperscript{20}. The final and ultimate goal will be the development of a fully automated and more accurate IBM algorithm for more generally shaped object.

ACKNOWLEDGEMENT

This work was supported by Electronics and Telecommunications Research Institute (ETRI) under contract 0701-2005-0064 and Inha University.

References


**Authors’ biographies:**

**Man Hee Lee** received his B.S. degree in Computer Engineering from Inha University in 1999. He is currently working toward the M.S. degree in the Information and Communication Engineering in Inha University. Since April 2007, he has been visiting Electronics and Telecommunications Research Institute (ETRI) as a visiting graduate researcher. His research interests include image-based modeling and rendering, sketch-based interface, GPGPU, and 3D game technology. He is a student member of ACM.

**In Kyu Park** received his B.S., M.S., and Ph.D. degrees from Seoul National University (SNU) 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 at Samsung Advanced Institute of Technology (SAIT). Since March 2004, he has been with the School of Information and Communication Engineering, Inha University, where he is an Assistant Professor. Since January 2007, he has been visiting Mitsubishi Electric Research Laboratories (MERL) as a visiting scholar. Current research interests include the joint area of computer graphics and vision, including image-based modeling and rendering, 3D face modeling, computational photography, and GPGPU. He is a member of IEEE and ACM.