PAPER *Special Section on Cyberworlds*

A Deformable Fast Computation Elastic Model Based on Element Reduction and Reconstruction

Shinya MIYAZAKI[†a)], **Mamoru ENDO**[†b)], ***Members***, **Masashi YAMADA**[†c)], ***Nonmember***, **Junichi HASEGAWA**[††d)], **Takami YASUDA**[†††e)], ***and*** **Shigeki YOKOI**[††††f)], ***Members***

SUMMARY This paper presents a deformable fast computation elastic model for real-time processing applications. 'Gradational element resolution model' is introduced with fewer elements for fast computation, in which small elements are laid around the object surface and large elements are laid in the center of the object. Elastic element layout is changed dynamically according to the deformation of cutting or tearing objects. The element reconstruction procedure is applied little by little for each step of the recursive motion generation process to avoid an increase in motion computation time.

key words: *elastic object, deformation, acceleration, element reduction, real time*

1. Introduction

Physically-based modeling is an effective way to generate natural motion of deformable objects by defining a comparatively simple model that represents the local properties of objects [1]. In the field of computer animation, it has been developed for the purpose of generating real motion of deformation as an animation. Thanks to great advances in computation performance, the possibilities are increasing more and more to extend real-time applications with high-resolution object models. At present more computation power is still needed. One possible solution is to develop a model with a small number of elements.

We propose a deformable fast computation elastic model. An original elasticity element model performs consistent restoration, and is applicable to any shape of polyhedron elements. To save motion computation time, variable sizes of elastic elements are laid gradationally in constructing elastic objects. Small elements are laid on the object's surface to keep high resolution. On the other hand, large elements are laid in the center of the object to save computation time, in which elements are invisible and transformation degrees are comparatively smaller than those on the object's surface.

Elastic models are often asked of function possibilities of being cut to be applied to practical cases such as surgical simulation. Our proposed model also realizes the function of cutting objects. In our model, cutting interaction makes large elements appear on the surface. To keep resolution on the new surfaces, the large elements are replaced with smaller ones. This reconstruction procedure needs more computation time. To limit the increment of computation time, all the elements are scanned only twice in every update cycle of motion generation. Although, the reconstruction procedure of an update cycle is limited, it is applied recursively during the motion generation process and proper results are generated.

1.1 Related Work

One of the earliest studies for deconstruction of elastic models is Ref. [2]. A deconstruction model was applied to fractured animations. Our deconstruction algorism study is based on it. Deconstruction models have been applied to practical application such as in surgical simulation [3]–[5]. Their interest is mainly in reconstruction of polygon-based models according to object deconstruction. Complex algorisms are necessary for them. Our model is a voxel-based one, and deconstruction is performed only by administrating element connections. It is introduced as being easy to handle and easily applied to general use. We made improvements in its motion computation acceleration.

2. Gradational Resolution Model

In this section, volumetric elasticity element model and gradational element resolution model are explained as fundamental models.

2.1 Volumetric Elasticity Elements

We have developed an original elastic element model. Either mass-and-spring model or simplex element in FEM (Finite Element Method) is generally used as an elastic element model [3]. Our model solves instabilities and improprieties in mass-and-spring model when elastic elements are strongly deformed. Our model's concept is simple and we

Manuscript received August 27, 2004.
Manuscript revised January 13, 2005.

[†]The authors are with the School of Computer and Cognitive Sciences, Chukyo University, Toyota-shi, 470–0393 Japan.
[††]The author is with the School of Life System Science and Technology, Chukyo University, Toyota-shi, 470–0393 Japan.
[†††]The authors are with the Graduate School of Information Science, Nagoya University, Nagoya-shi, 464–8601 Japan.
a) E-mail: miyazaki@sccs.chukyo-u.ac.jp
b) E-mail: endoh@sccs.chukyo-u.ac.jp
c) E-mail: myamada@sccs.chukyo-u.ac.jp
d) E-mail: hasegawa@life.chukyo-u.ac.jp
e) E-mail: takami@nagoya-u.jp
f) E-mail: yokoi@is.nagoya-u.ac.jp

DOI: 10.1093/ietisy/e88–d.5.822

have developed a fast computation method to solve it numerically [6]. It permits any number of vertices in the element. It is an advantage to be free from subdividing elements into simplex elements or truss structured springs, which is necessary in other models. Free vertex layout is a necessity for constructing our gradational resolution model (GRM).

2.2 Definition of Elastic Force

Followings are definition of elastic force in our element model. Restoration force is proportional to the vertex displacements in the element from the reference position, which gives its reference shape, the goal of element restoration (Fig. 1). The reference shape's location is decided in each element as is; both the resultant internal force and the moment of force of the internal stresses are equal to zero vectors.

$$\sum_i \boldsymbol{r}_i \times k(\boldsymbol{R}_i - \mathbf{r}_i) = k \sum_i \boldsymbol{r}_i \times \boldsymbol{R}_i = \vec{0} \tag{1}$$

Here, vector $\boldsymbol{R}_i$ and $\boldsymbol{r}_i$ are relative positions of vertex i in the reference shape and the deformed shape with respect to the center of gravity, respectively. Force is proportional to displacements with proportionality constant k. In the implementation, the initial values of $\{ \boldsymbol{R}_i \}$ are registered as $\{ \boldsymbol{Ro}_i \}$, and $\{ \boldsymbol{R}_i \}$ is defined as a result of the rotation of $\{ \boldsymbol{Ro}_i \}$ around the center of gravity.

Our model becomes relatively similar to FEM model in elastic force definition when element shapes are of a regular triangle shape or regular tetrahedron shape. After elastic force is decided, velocity and position for each vertex is renewed by difference equations of motion that are established locally in each vertex.

2.3 Policy for Element Reduction

One simple method of element reduction is by replacing a few elements with one large element. If the element reduction procedure is performed and applied, only when element deformations are comparatively small, this could be an effective element reduction. More computation time is needed to search such elements. Also, it is difficult to predict deformation tendencies beforehand. However, deformation is large around contact points with rigid objects. External force from rigid objects is far larger than internal force in elastic objects, and works directly on the surface of elastic objects. Connection between elements also constrains elements' deformation. This works less on the surface where fewer neighboring elements are joined.

Based on the above considerations, we proposed a model whose element size can be gradationally increased from the surface to the center (Fig. 2). The minimum element's shape is square and has one vertex on each corner. The other elements' shapes are more complex. It effectively saves computation time and maintains the original voxel resolution on the surface. Figure 3 shows differences between our GRM and the ordinal voxel model. Figures 3 (a) and (c) are captured figures during interactive manipulation. The difference in outline shape is comparatively small with close consistencies.

Computation's size is roughly proportional to the number of elements. A n by n by n cubic voxel object has n^3 elements. They are reduced into $8n^2$ by the gradational resolution model, which has the same resolution on the surface. Compression rate is about $8/n$. Compression becomes more effective in large numbers of n. In two dimensional cases in

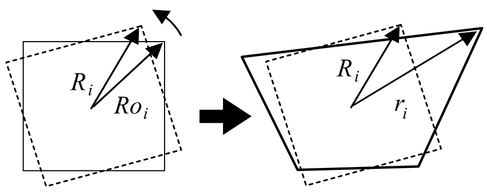


Fig. 1 Vertex displacements in a volumetric elasticity element. Solid lines show the original reference shape, broken lines show the rotated reference shape, and bold lines show the deformed shape.

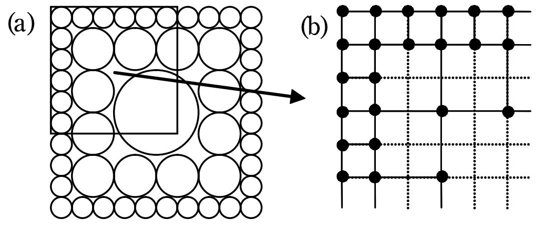


Fig. 2 Image illustrations of a gradational element resolution model. (b) is a magnified illustration of a corner of a cubic object (a).

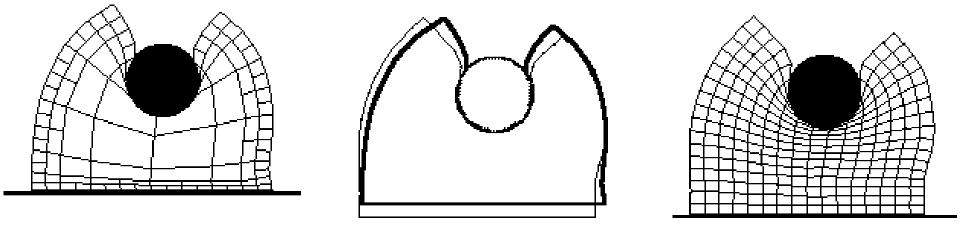


Fig. 3 Comparing shapes between GRM and the ordinal voxel model during interactive manipulation. In figure (b), the thick lines and the thin lines show the outlines of GRM and the voxel model, respectively.

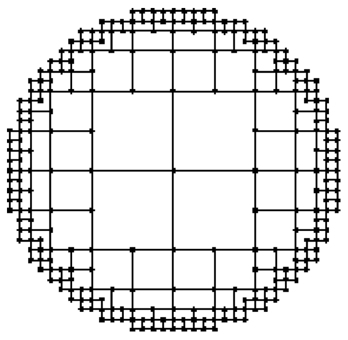


Fig. 4 Element and vertex layout of gradational resolution model for a round shape object.

Fig. 4, a round object, which can fit into a 32×32 square, has 852 square elements before reduction. After reduction, it is constructed by 172 elements. 79.8% reduction in processing time was expected due to element reduction. However, in real simulation, 86.3% reduction was measured under the condition of keeping 30 fps drawings, because more simulation cycles can be executed to a drawing cycle after element reduction.

3. Element Reconstruction for Deformation

This section explains element reconstruction procedures according to object deformation in two dimensional cases.

3.1 Element Look Up Table

Complex programming is necessary to modify gradational resolution models. Also, vertex sharing among elements must be administrated during the deformation process. An element LUT (look up table) is introduced to simplify the reconstruction procedure. Figure 5 gives an outline of the LUT. The LUT consists of multi layers which correspond to each element resolution in the gradational resolution model. By accessing element information through the LUT, element connectivity or shared vertices between elements are easily checked. For example, in Fig. 5, all the elements that includes either the vertex on level 1 (2, 2) or the vertex on level 2 (1, 1) are selected as the elements connected at that vertex position.

LUT is maintained not to overlap elements among layers during the reconstruction procedure. Level differences between two neighboring elements must be 0 or 1. Keeping these conditions in programming prevents us from bugs.

3.2 Reconstruction Procedure

The cutting operation applies external force to separate elements from each other. Element separation is achieved by releasing sharing vertices when external force beyond a certain threshold value is applied. During the cutting operation, element separation is propagated from the object's surface to the center. As a result, a cracking phenomenon is formed. In the gradational model, internal large elements appear on the object surface. Those elements should be divided into smaller ones in order to maintain the element gradation. If each element is replaced with quarter elements only once for each update cycle of motion computation, proper results can be achieved, because the reconstruction process is recursively applied during the motion generation process.

Figure 6 illustrates a process of element reconstruction. Reconstruction procedures are advanced as follows.

[phase 1]
a) Generating middle points in the original element.
b) Generating quarter elements.

[phase 2]
c) Removing unused middle points.
d) Reconstruction of the original element.

Here, phase 1 is applied in order to every element according to size increment before applying phase 2. After that, phase 2 is applied to every element in the same way. As a result, every element is scanned twice.

In phase 1, each element is checked to see if its quarter elements should be generated or not, by the conditions shown in the next section. In generating quarter elements, some middle points have to be generated. Procedure a) of phase 1 generates all middle points temporarily for easy programming.

In phase 2, the temporarily generated middle points in phase 1 which are not being used for quarter elements are removed. Then, the original element is reconstructed and laid over the other quarter elements. Here, elastic force has to be recalculated according to element generation and reconstruction.

3.3 Quarter Element Generation

The condition for quarter element generation differs between level 1 and level 2 and up. Figure 7 (a) shows the case of level 1. Three vertices, that are the vertex of the quarter element and neighboring level 0 elements, are checked. If either of them becomes a released vertex by element separation, the quarter element is generated, because it appears on the object surface (Fig. 8 (c)).

Figure 7 (b) shows the case of level 2 and up. It is checked to see if two lower level's elements neighbor the quarter element. If any of the four possible elements are found, the quarter element is generated to maintain element gradation. All of level $n - 2$ elements must be processed before the reconstruction process for level n elements. It is guaranteed because the order of level increment is kept in the reconstruction procedure.

3.4 Implementation

A two dimensional manipulation environment is implemented (Fig. 9). A rigid circular manipulator is controlled by a mouse. Collision force is applied to element vertices to

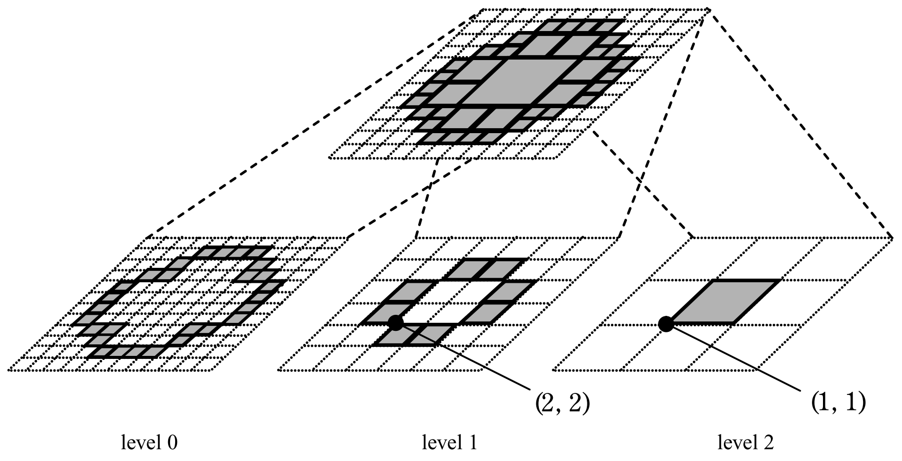


Fig. 5 Multi-layered element look up table for size.

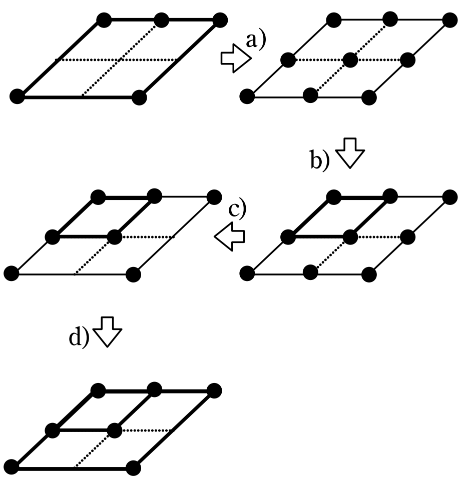


Fig. 6 An example of element reconstruction.

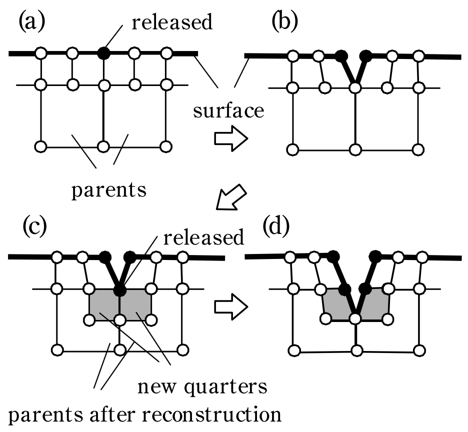


Fig. 8 Generation of quarter elements in the reconstruction procedure.

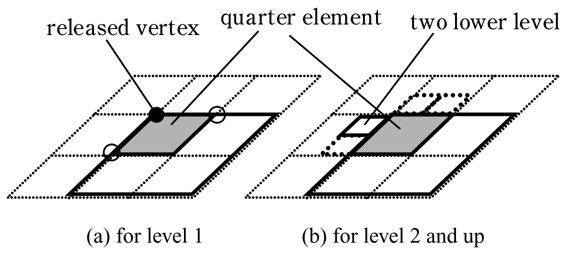


Fig. 7 Conditions for generating quarter elements in the reconstruction procedure.

push them outside the manipulator when the vertices are inside the manipulator. Element separation is caused by strong collisions, and then element reconstruction is executed according to them. Collisions between elastic elements are not detected at this point.

We have also developed a three dimensional model (Fig. 10). Program codes are downloadable from our web site [7].

3.5 Performance Results

Object-oriented programming really contributes to writing easily understood program codes. Especially, a must for sharing codes in a group. The downside to this is it decreases computation performance. Our program is written in C++ in which priority was given over fast computation to easily understanding (OOP version) codes. Another version is derived from the OOP version for fast computation by expanding hierarchic function calls that is an essential reason for decreasing performance. The fast computation version was derived from the OOP version. The derived class is also provided in the same program [7]. Then, class variables are used only as structure variables. Also, declarations of local class variables are removed. For example, a class variable for a three dimensional vector is replaced with three double variables.

Table 1 shows an increase in performance. In the Intel Pentium 4 HT 2.8 GHz, computation performance had become about three times better by rewriting program codes. Especially, it is effective in the velocity-and-position update process which consists of very simple procedures. Compare

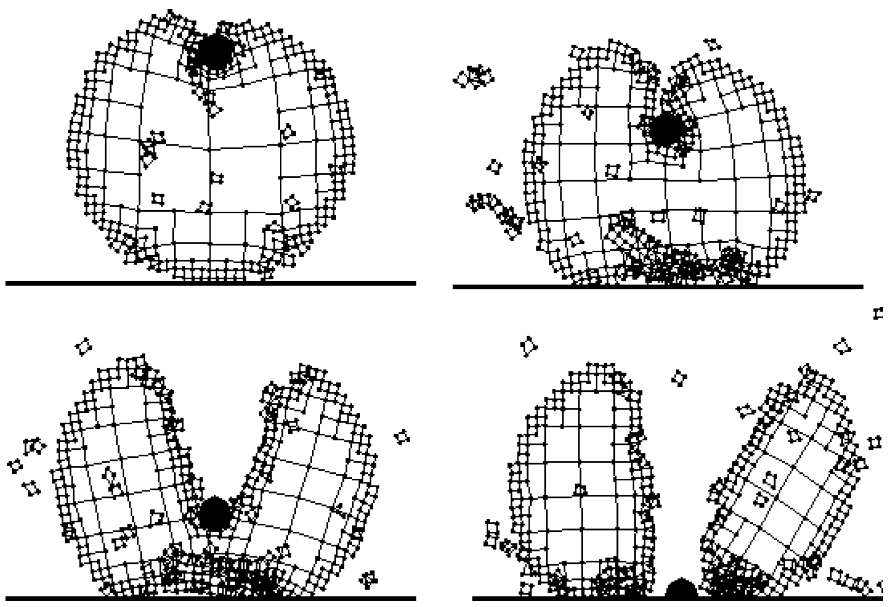


Fig. 9 Reconstruction process according to cutting operation.

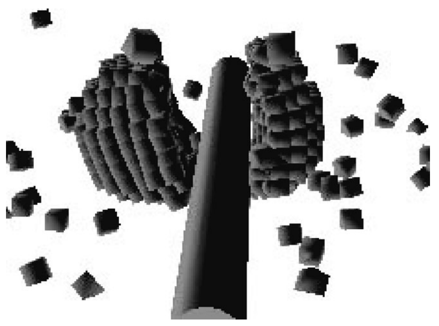


Fig. 10 An example of three dimensional model. Minimum cubic elements are shaded as a sphere for clarification.

Table 1 Computation times for dynamic simulation. They are ones per element and averaged at ten thousand times. Created codes are optimized for maximum speed.

(μs)

	Pentium4 (2.8GHz) OOP version	Pentium4 (2.8GHz)	Athlon64 3200++ (2.2GHz)
a	0.52	0.26	0.16
b	0.33	0.23	0.24
c	0.15	0.12	0.05
d	0.94	0.08	0.06
e	1.39	0.38	0.26
total	3.33	1.07	0.78

a: elastic force, b: reconstruction, c: force summation, d: velocity-and-position update, e: collision

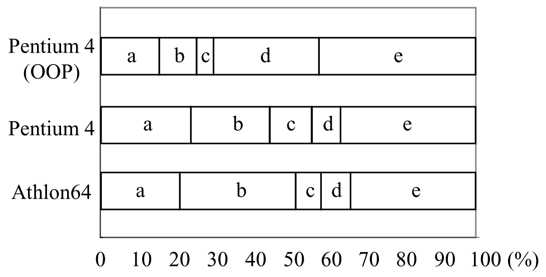


Fig. 11 Percentages of processes in computation time.

the upper and middle lines in Fig. 11.

We also had a chance to measure in a 64-bit processor AMD Athlon64 3200++ whose real clock speed is 2.2 GHz. Microsoft Windows XP 64-bit Edition Beta and Visual Studio 2005 Beta are now available. It accelerates in the processes of elastic force calculation, force summation, and collision, which are realized by simple procedures compared with the reconstruction process. Compare the middle and lower lines in Fig. 11.

4. Conclusions

We proposed a fast computation elastic model. In the gradational element resolution model, smart algorithms have been developed for dynamic element layout reconstruction according to object deformation. Our model is directed for wide use, for it is a voxel-based one and supports a cutting operation. In future works, it will be applied to practical applications such as real-time interaction with dynamic behavior of soft tissue in surgical simulation.

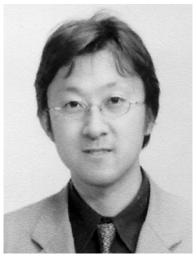
Acknowledgements

We would like to thank Professor Teruo Fukumura of

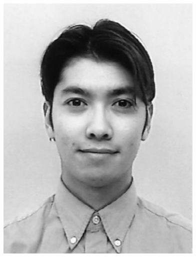
Chukyo University for his encouragement and assistance. Special thanks to Marvin Lenoar, a great English teacher and for English assistance on this project. This research was supported in part by the grant-in-aid for Private University High-Tech Research Center and grant-in-aid for Young Scientists from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

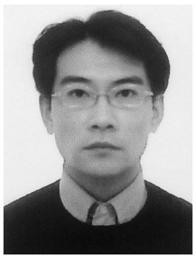
[1] D. Terzopoulos, J. Platt, A. Barr, and K. Fleisher, “Elastically deformable models,” Comput. Graph., vol.21, no.4, pp.205–214, 1987.
[2] A. Norton, G. Turk, B. Bacon, J. Gerth, and P. Sweeney, “Animation of fracture by physical modeling,” Vis. Comput., vol.7, no.4, pp.210–219, 1991.
[3] S. Cotin, H. Delingette, and N. Ayache, “A hybrid elastic model for real-time cutting, deformations, and force feedback for surgery training and simulation,” Vis. Comput. vol.16, no.7, pp.437–452, 2000.
[4] D. Bielser and M.H. Gross, “Interactive simulation of surgical cuts,” Pacific Graphics 2000, pp.116–125, 2000.
[5] H. Nienhuys and A. Stappen, “A surgery simulation supporting cuts and finite element deformation,” MICCAI 2001, pp.145–152, 2001.
[6] S. Miyazaki, J. Hasegawa, T. Yasuda, and S. Yokoi, “Acceleration of elastic model’s motion computation based on elastic element reduction,” CG International 2002, pp.239–246, 2002.
[7] URL: http://www.om.sccs.chukyo-u.ac.jp/



Shinya Miyazaki is an associate professor at Chukyo University, School of Computer and Cognitive Sciences. His research interests are in application of computer graphics and virtual reality. He received a B.E., M.E., and Ph.D. degree in information engineering from Nagoya University.



Mamoru Endo is a lecturer at Chukyo University, School of Computer and Cognitive Sciences. His research interests include network systems, computer graphics, virtual reality and their application in real society. He received a B.E. degree in information engineering from Shinshu University, and a M.E. and Ph.D. degree in human informatics from Nagoya University.



Masashi Yamada is a lecturer at Chukyo University, School of Computer and Cognitive Sciences. His research interests are in application of artificial intelligence to the computer graphics and multimedia systems. He received a B.E. and M.E. degree in electrical and computer engineering, and a Ph.D. degree in engineering from Nagoya Institute of Technology.

Junichi Hasegawa is a professor at Chukyo University, School of Life System Science and Technology. His research interests are in application of image processing and computer vision to the field of medicine and sport. He received a B.E. degree in electrical engineering, and a M.E. and Ph.D. degree in information engineering from Nagoya University.

Takami Yasuda is a professor in the Graduate School of Information Science at Nagoya University. His research interests are in application of computer graphics and virtual reality. He received a B.E. and M.E. degree in electronic engineering from Mie University, and a Ph.D. degree in information engineering from Nagoya University.

Shigeki Yokoi is a professor in the Graduate School of Information Science at Nagoya University. His research interests are in application of compute graphics and virtual reality to social and multimedia systems. He received a B.E., M.E., and Ph.D. degree in electrical engineering from Nagoya University.