An Alternative To The Hemicube Algorithm For Computing Form factor

Sungye Kim, Kyunghyun Yoon

Department of Image Engineering,

Graduate School of Advanced Imaging Science, Multimedia, and Film,

Chung-Ang University, Seoul, South Korea

Abstract

The radiosity method is one of the methods used in global illumination simulation, and it is a rendering technique that requires a lot of CPU time. The form factor calculation is the most time consuming part of its flow. Since hemisphere was introduced in the radiosity algorithm, it has been the most basic form factor calculation model. In this paper, we use a geodesic dome to approximate the hemisphere, and represent the dome with a set of points. When the form factor is calculated, a *kD* Tree can be also applied to select the projected region. This method is simpler and faster than the conventional hemicube approach, because it avoids processing such as polygon clipping.

Keywords: *form factor, radiosity, hemicube, spatial subdivision*

1. INTRODUCTION

Global illumination is essential for realistic rendering. In global illumination, radiosity simulate the propagation of light throughout the scene very well. Radiosity produces a more realistic image by capturing the diffuse reflection of light. Radiosity was introduced by Goral et al.[8] and Nishita et al.[9], and it has been studied to simulate the diffuse reflected light. Radiosity manages diffuse reflection of light in the enclosure. Consequently, soft shadows and more natural color bleeding effects are possible. One of the advantages of radiosity is its view independent property. Because a form factor is a strictly geometric quantity, its value depends only on the shape and relative location of surfaces in the scene. This property enhances the speed of walkthrough because it doesn't require the form factor to be re-calculated under different illumination conditions. The computation of form factor is generally the most time-consuming part of the radiosity algorithm. Therefore, fast and accurate calculation of the form factor is one of the main ways to improve the radiosity algorithm. A new algorithm to speed up the form factor calculation is proposed in this paper, and its result is compared with those of previous algorithms.

2. RELATED WORKS

Form factor calculation is a very important but time consuming step when determining a radiosity solution. Calculating the form factor takes up to 90% of the time of a radiosity flow[1]. So, it is important to calculate the form factor more rapidly. The form factor is the physical property that represents heat transfer between two patches. Form factor, F_{ij}, is represented by the ratio of the two energies, one of which is radiated from the Lambertian surface, E_i, the other of which is absorbed into patch, E_j. The reflectance or emittance properties of two patches don't influence the setting of the form factor because the form factor depends only on the geometry and the orientation of the two patches. The analytical expression of the form factor is given by (1), where function H_{ij} represents the visibility between the patches E_i and E_j[4].

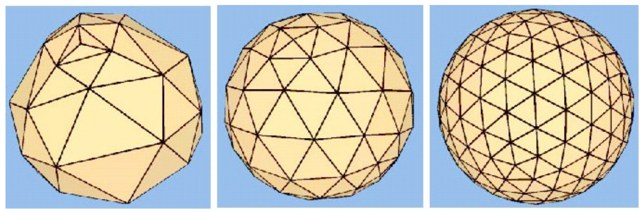
$$F_{dAij} = \frac{1}{A_i}\int_{Ai}\int_{Aj}\frac{\cos\theta_i \cos\theta_j}{\pi r^2} H_{ij}\, dA_j\, dA_i \qquad (1)$$

- A_i : An area of the patch_i
- A_j : An area of the patch_j
- r : A length of the ray between the differential areas dA_i and dA_j
- H_{ij} : Visibility function
- θ_i : An angle that the ray makes with patch_i's normal
- θ_j : An angle that the ray makes with patch_j's normal

The form factor calculation algorithms can be divided into an analytic method[10], a ray tracing method[11][12] and a projection method using hemi-space[5][13]. The analytic method uses a contour integral to calculate the form factor. Goral[9] used a contour integral using the Stoke's law. However, this method can only be guaranteed for environments with no occlusion. The algorithm using the ray tracing method[11], introduced by Wallace, uses sampling to calculate the form factor. Although this method can be applied to curved environments, it takes a large amount of processing time. The two methods mentioned above guarantee the accuracy of the calculation, but they use a lot of processing time and are therefore not suitable to apply to a real time application. Many different methods using the hemi-space have been suggested and each of them uses mainly the hemicube, cubic tetrahedron, and hemisphere. To subdivide the hemisphere, Spencer[14] used the traditional spherical polar split. But he immediately rejected the method because of the wildly varying areas of the surface elements of the hemisphere. Subsequently, he found equi-area regions by using unevenly spaced values of Φ in the discretization. Foley et al.[15] used a modified method that had originally been proposed by Renka[18]. Renka's algorithm shows that the surface of a unit sphere is approximated with a triangular mesh of points based on the regular tetrahedron. This is one of the platonic solids. Renka created a discretized hemisphere by finding the midpoint of each of the geodesic arcs that correspond to the tetrahedron's edges. But Foley used an alternative method of the Renka's algorithm, because it had a disadvantage in that it doesn't allow the user to select the number of split triangles. For example, if one requires a triangle to be split into 3,000 triangles, then the specific options are either 1,024 or 4,096. Neil Gatenby et al.[3] introduced an alternative to the hemicube algorithm that used a unit hemisphere as its view volume. To make a hemisphere, they used non-platonic tetrahedron, because it results in a less regular distribution of areas than would have been achieved with the platonic tetrahedron suggested by Foley et al. In their algorithm, each direction formed by the discretization was considered the form factor. They used a Renka's algorithm and Sillion et al.[17]'s method to calculate the form factor. Eric Languenou et al.[7] suggested a method of form factor calculation combining hemisphere and ray tracing. In their paper, the hemisphere is discretized into surface elements by the spherical coordinate, and each element corresponds to a small form factor called delta form factor. This method requires the N×M intersection test, where N is the number of discretized elements of the hemisphere and M is the number of elements within the environment. It fires a ray of light from the center of the hemisphere to the selected points on the surface of the hemisphere. Consequently, the number of pixels on the hemisphere greatly affects the quality of the final image. However, the accelerating method from the ray tracing must be included to improve the speed in a complex environment. Eric Languenou et al. used the acceleration method using spatial subdivision and a subdivision tree to overcome this problem. Akio doi and Takayukiitoh[2] proposed an improved version of the methods suggested by S. N. Spencer[14] and Van Wyk Jr.[16]. Spencer and Van Wyk proposed hemisphere base methods with edge subdivision. They calculated the degree of the arc between pairs of points projected onto a hemisphere base. This value was used to determine a number of intermediate points along the element's edge. However, the method using only edge subdivision failed to remove hidden surfaces. Their method subdivided patches in the scene into small triangles before projecting them onto the hemisphere. The form factor was calculated by projecting each triangle onto the base of the hemisphere. Akio also used a parallel processor that divided the base of the hemisphere into regions and assigned a region to each processor to obtain a high performance. Vincent Jolivet[19] presented the improvements that solved the hemisphere subdivision method’s main drawback is a very important cost in memory occupation.

3. FORM FACTOR CALCULATION

Figure 1 shows the overall implementation flow of the system. Initially, information about 3D model and geodesic dome is created. Next, configure *kD* tree using information about the dome from the preprocessing step. And then, among the patches in the environment, select the one that has the highest energy and set it as a energy emitting patch. That is processed in projection routine. The form factor is calculated between the patch chosen and all the other patches. Because we use progressive refinement method, we can create intermediate images at each step. After each step arrives at a satisfied convergence, we can get result images.



(a) by tetrahedron (b) by octahedron (c) by icosahedron

Figure 2: Geodesic domes

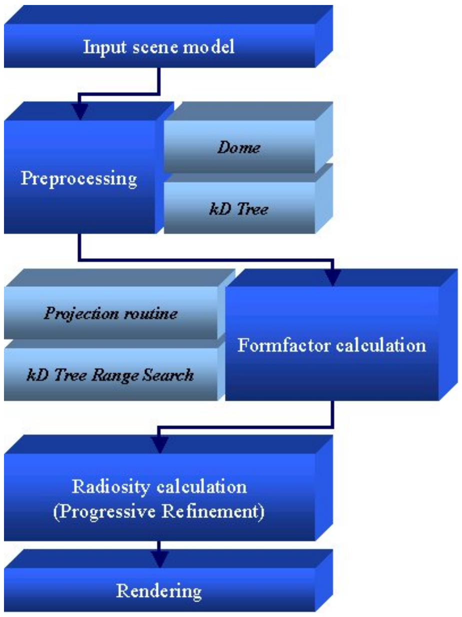
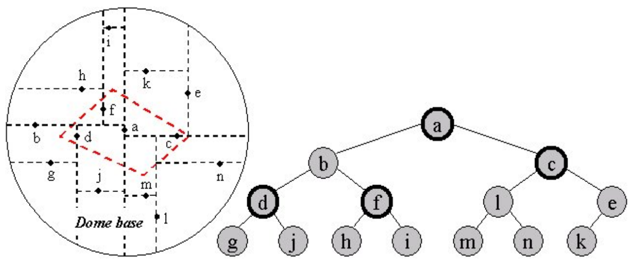


Figure 1: Implementation flow

The word "geodesic" means to divide the shape of a sphere such as the earth into exact points. Generally, a geodesic dome is a hemisphere composed of a lot of triangles. Various types of geodesic domes can be created according to the initial model. Tetrahedron, octahedron and icosahedron are used mainly as a base model. Figure 2 shows geodesic domes being created with various basic units. The icosahedron approximates a sphere most closely as shown in figure 2. We can consider the edge of a patch projected on the dome as a collection of straight-line segments.

3.1 kD Tree

The factor, k in the 'kD tree' denotes the dimensional view of the tree. That is to say, if we use 2 dimensional coordinates then, the kD tree becomes a 2D tree. 2D tree is a binary tree which recursively subdivides the plane by horizontal and vertical cut lines[6]. As it descends down the tree, the cut lines alternate in direction. Even levels contain vertical cut lines, and odd levels contain horizontal cut lines. This suggests that a set of points on a 2 dimensional plane is a set S, then a horizontal or vertical line which goes through a point in the S set becomes a value of a node in the tree. Figure 3(a) and (b) represent a subdivision by kD tree algorithm and the tree corresponding to it. Figure 3(b) shows the nodes, □, □, □, □, that are points contained in a rectangle query range like figure 3(a).



(a) A subdivision of the dome base (b) kD Tree

Figure 3: a kD Tree structure

2D tree is constructed over a set of n points in T(n) time, where T(n) is expressed by the recurrence, (2), for suitable constants a and b. Hence $T(n) \in O(n\log n)$.

$$T(n) = \begin{cases} 2T(\frac{n}{2}) + an & \text{if} \quad n > 1 \\ b & \text{if} \quad n = 1 \end{cases} \tag{2}$$

3.2 Form factor calculation using the Geodesic dome

In this paper, we use an icosahedron as a basic unit to form the geodesic dome. The dome that is used in our system is created with 3DS Max R2 of Kinetic. As a result, a frequency of the dome is static, and we must construct data structure of the dome in preprocessing. To calculate form factor, we use a delta form factor notion of hemicube algorithm. And to simplify clipping process, we apply a kD tree to determine the projected region. In preprocessing, the geodesic dome is moved to the center of the emitter. This emitter is a patch that has the highest energy. For calculating form factor of patches in the environment and the emitter, each patch projects to the center of the dome. Form factor calculation of each dome cell in the projection region moves on to the form factor calculation of patches in the environments and the emitter. That is, each cell of the geodesic dome corresponds to a grid cell of the hemicube. Delta form factor of each cell can be easily obtained by examining the cells in the projected region because flat triangles cover the surface of the dome.

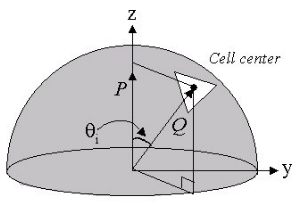


Figure 4: Form factor geometry for cell of the dome

In this paper, form factor of each cell in the dome is represented in (7) which combined equations (3) to (6).

$$P = \text{emitter .normal} \tag{3}$$

$$Q = \text{domecell .center} - \text{emitter .center} \tag{4}$$

$$\cos\theta_i = \frac{P \cdot Q}{|P||Q|} \tag{5}$$

$$\cos\theta_j = \frac{(-Q)\cdot(-\text{domecell .normal})}{|-Q||-\text{domecell .normal}|} \approx 1 \tag{6}$$

$$F_{dEiEj} \approx \frac{\cos\theta_i \cos\theta_j}{\pi(\text{radius}_{\text{dome}})^2}\text{Area}_{\text{cell}} \tag{7}$$

$$= \frac{\cos\theta_i}{\pi}\text{Area}_{\text{cell}}$$

$$= \frac{1}{\pi}\frac{P \cdot Q}{|P||Q|}\text{Area}_{\text{cell}}$$

• emitter	An emitter patch radiating energy
• domecell	A cell of dome located in projected area
• P	A normal vector of emitter
• Q	Vector from center of emitter to center of dome cell
• $\text{radius}_{\text{dome}}$	A radius of the dome
• $\text{Area}_{\text{cell}}$	An area of a cell

The value of $\cos\theta_j$ in equation (6) is near 1, because the angle between the normal vector of each dome cell, P, and vector Q to the center of each dome cell is almost 0 degree. Finally, form factor of each cell within the dome is represented by (7). The radius of the dome is 1, and the area of the each cell is easily determined by calculating the area of the cell.

3.3 Determining cells in the projected region

The kD tree is used for performing a range search about the projected region from the base of the geodesic dome. Typical kD tree search uses a rectangle area for searching, but in this paper, we applied an arbitrary quadrilateral to search a query range. Points shown in figure 5 are represented in 2D coordinates,(x, y) by projecting the center coordinates, which is representative of each cell consisting of a surface that is on the base of the dome.

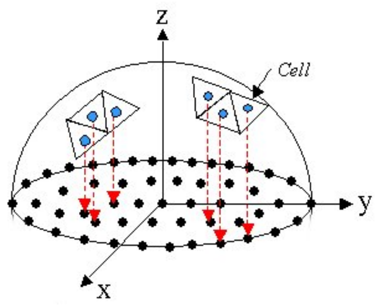
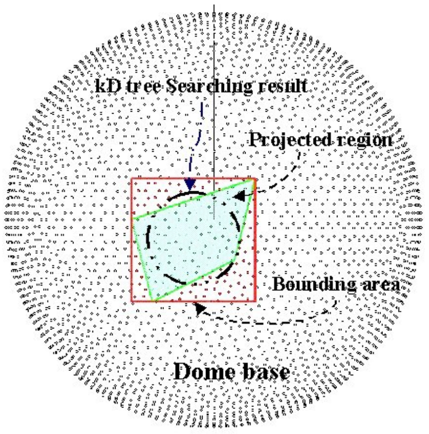
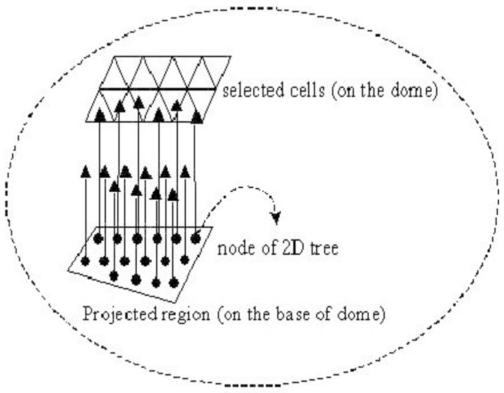


Figure 5: 2D representation of a Dome

When an arbitrary shaped polygon is given as a query range, a minimum sized bounding rectangle surrounding the polygon is created. Show figure 6(a). Candidate points for the range are selected by typical kD tree search method which is based on the bounding rectangle. As shown in figure 6(a), we select points included in a projection field of real element among all of the selected candidate points. Using the information about the searching result in figure 6(a), we can select dome surface cell as figure 6(b). As a result, selected cells on the dome surface are included in an inner range created by projection onto dome surface. In conclusion, we determine the total sum of delta form factor of those cells between the emitter and the element.




selected cells (on the dome)
node of 2D tree
Projected region (on the base of dome)

(a) Top (b) Bottom

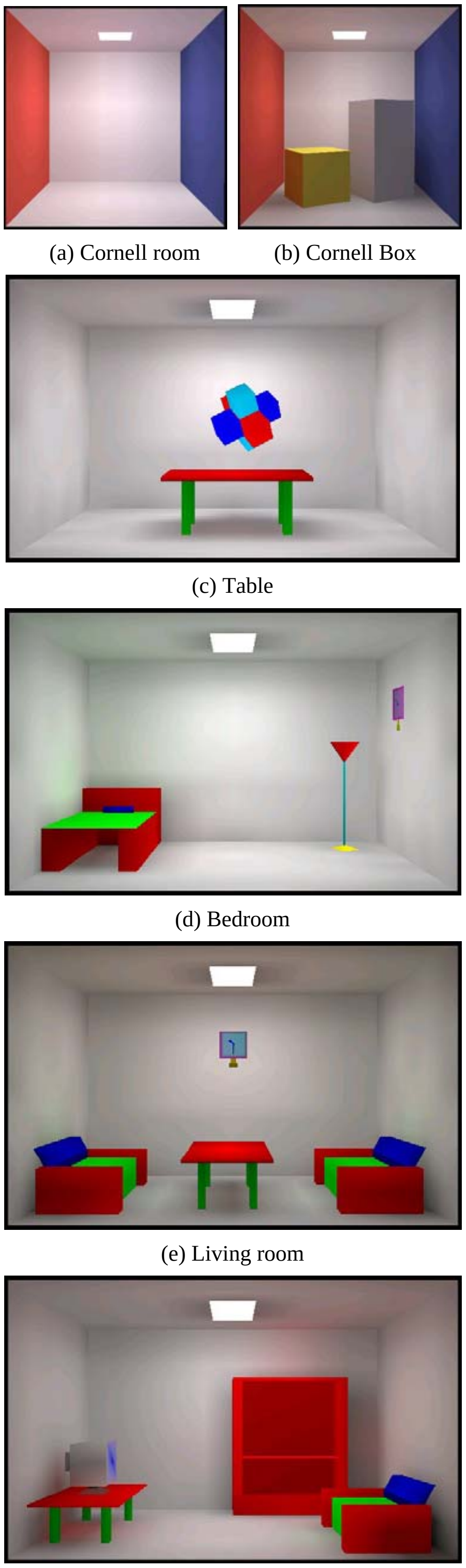
Figure 6: Determining cells in the projected region

4. CONCLUSIONS AND FUTURE WORKS

Figure 7 shows the images generated by the proposed method. To evaluate the results, we measured elapsed time between hemicube algorithm and proposed geodesic algorithm according to the models given by Table 1. Figure 8 shows the comparative results over an elapsed time for hemicube and geodesic methods, using test models. The resolution of the hemicube is 400. In the figure 8, the horizontal axis shows the increasing complexity of the models, and the vertical axis indicates the elapsed time ratio for the test. We set the elapsed time by hemicube method to 1. The graph shows that the geodesic method consumes less time than the hemicube method. With convergence obtained through processing of the hemicube method whose iteration step is 100, we measured the elapse time of a proposed algorithm until the distribution of energy comes close to the convergence.

Table 1: Information of test models

	Instance	Surface	Patch	Element
Cornell room	7	7	127	151
Cornell Box	9	10	137	191
Table	10	18	193	681
Bedroom	10	18	193	681
Living room	11	19	221	775
TV and sofa	11	15	237	806



(a) Cornell room (b) Cornell Box

(c) Table

(d) Bedroom

(e) Living room

(f) TV and sofa

Figure 7: Result Images

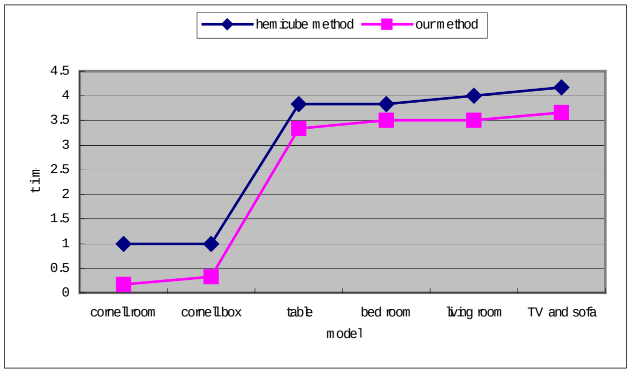


Figure 8: Elapsed time comparison for each model

The current radiosity rendering system demands fast processing of complex environments. To meet such demand, a fast calculation of the form factor is required. The method described in this paper calculates the form factor by projecting the elements in an environment on the dome. The dome is composed of many triangularly shaped cells that are almost equal in area. The form factor of the elements can be obtained by determining the cells within the region of the projected element. We used a kD search tree for selecting the cells of the dome contained in projection area of each element. With further research, accuracy of form factor can be tested for more various conditions, geodesic adaptive subdivision algorithm can be applied to create a dome implicitly for the complex environment and also hardware support can be considered.

5. REFERENCES

[1] G. Pietrek, *Fast Calculation of Accurate formfactors*, Fourth Eurographics Workshop on Rendering, pp.201~220, June, 1993.

[2] A. Doi, I. Takayukiitoh, *Acceleration Radiosity Solution Through the Use of Hemisphere-base formfactor Calculation*, The Journal of Visualization and Computer Animation, 9(1): 3~15, 1998.

[3] Neil Gatenby, T. Hewitt, *Radiosity in Computer Graphics: A Proposed Alternative to the Hemi-cube Algorithm*, Second Eurographics Workshop on Rendering, April, 1991.

[4] l. Ashdown, *Radiosity: A Programmer's Perspective*, John Wiley & Sons, Inc., 1994.

[5] Beran Koehn, Pavicic, *A Cubic Tetrahedral adaptation of the Hemicube Algorithm*, pp. 299~302, 1991.

[6] M. J. Laszlo, Computational Geometry and Computer Graphics in C++, Prentice Hall, 1996.

[7] Eric Languenou et al., *An adaptive Discretization Method for Radiosity*, Eurographics 11(3): C-205~216, 1992.

[8] C. M. Goral, K. E. Torrance, D. P. Greenberg, and Bennett Battaile, *Modelling the Interaction of Light Between Diffuse Surfaces*, Computer Graphics (SIGGRAPH '84 Proceedings), 18(3): 212~222, July, 1984.

[9] Tomoyuki Nishita, and Eihachiro Nakamae, *Continuous tone representation of three-dimensional objects taking account of shadows and interreflection*, Computer Graphics (SIGGRAPH '85 Proceedings), 19(3): 23~30, July, 1985.

[10] D. R. Baum, H. E. Rushmeier, and J. M. Winget, *Improving radiosity solutions through the use of analytically determined form-factors*, Computer Graphics, 23(3): 325~334, 1989.

[11] J. R. Wallace, K. A. Elmquist and E. A. Haines, *A ray tracing algorithm for progressive radiosity*, Computer Graphics, 23(3): 315~324, 1989.

[12] Maxwell, Gregory M., Michael J. Bailey, Victor W. Goldschmidt, *Calculations of the Radiation Configuration Factor Using Ray Casting*, Computer-Aided Design, 18(8), pp. 371~379, Sept., 1986.

[13] M. F. Cohen, D. P. Greenberg, *The Hemicube Radiosity Solution for Complex Environments*, Computer Graphics (SIGGRAPH '85 Proceedings), 19(3): 31~40, Aug. 1985.

[14] S. N. Spencer, *The hemisphere radiosity method: a tale of two algorithms*, Photorealism in Computer Graphics, Springer-Verlag, pp.127~135, 1992.

[15] T. A. Foley, D. A. Lane, G. M. Nielson, and R. Ramaraj, *Visualizing functions over a shpere*, IEEE Computer Graphics & Application, pp.32~40, January, 1990.

[16] G. C. Van Wyk Jr., *A geometry-based insolation model for computer-aided design*, PhD thesis, The University of Michigan, 1988.

[17] Francois Sillion, Claude Puech, *A general two-pass method iterating specular and diffuse reflection*, ACM SIGGRAPH Computer Graphics, 23(3): 335~344, July, 1989.

[18] R. J. Renka, *Interpolation of data on the surface of a sphere*, ACM Transactions on Mathematical Software, 10(4): 417~436, December, 1984.

[19] Vincent Jolivet, Dimitri Plemenos, *A New Hemisphere Subdivision Technique For Computing Radiosity*, Proceedings of the 8th International Conference and Exhibition on Computer Graphics and Visualization in Russia, Graphicon'98, September, 1998.

Acknowledgement

This research was partially supported by the Brain Korea 21(BK 21) program of the Ministry of the Education.

About the author

Sungye Kim is the PhD student of Chung-Ang University of Applied Computer Graphics in Seoul, South Korea. She received a M.S in computer science from Chung-Ang Univ. in 2000. Her research interests include global illumination, radiosity, non-photorealistic rendering, and Image-based rendering.

E-mail: inside@cglab.cse.cau.ac.kr

Homepage: http://165.194.29.150

Telephone: +82-2-820-5308 / Fax: +82-2-820-5301

Postal Address: Dept. of Image Engineering, Graduate School of Advanced Imaging Science, Multimedia, and Film, Chung-Ang University, 221, Huksuk-Dong, DongJak-Ku, Seoul, Korea.

Kyunghyun Yoon is the Professor of Chung-Ang University of Applied Computer Graphics in Seoul, South Korea. He received a PhD in computer science from Univ. of Connecticut in 1991. His research interests include global illumination, non-photorealistic rendering, and Image-based rendering.

E-mail: khyoon@cglab.cse.cau.ac.kr