Fast Computation of the Visibility Region Using the Spherical Projection Method

Gil Whoan Chu and Myung Jin Chung

Abstract: To obtain visual information of a target object, a camera should be placed within the visibility region. As the visibility region is dependent on the relative position of the target object and the surrounding object, the position change of the surrounding object during a task requires recalculation of the visibility region. For a fast computation of the visibility region so as to modify the camera position to be located within the visibility region, we propose a spherical projection method. After being projected onto the sphere the visibility region is represented in θ-ϕ spaces of the spherical coordinates. The reduction of calculation space enables a fast modification of the camera location according to the motion of the surrounding objects so that the continuous observation of the target object during the task is possible.

Keywords: visibility region, sensor planning, spherical projection

I. Introduction

When the surrounding objects of a workspace change their position during a task, visual information would be a guide to planning and teaching robot motions for a successful accomplishment of the task. Therefore, visual feedback plays an important role in many fields such as object tracking, assembly, and inspection task [1][2][3]. To obtain visual information of a target object, the camera should modify its position and orientation according to the change of the surrounding objects' positions. Recently, there has been much research on sensor planning. The goal of this research is the automatic selection and modification of the camera configurations based on the priori geometric knowledge of the environment, such as CAD models [4][5][6][7].

For the visibility constraint, C. K. Cowan and P. D. Kovesi computed the three dimensional region from where a convex target object can be viewed entirely from a surrounding object [8][9]. The total occluded region is obtained as the union of the component occluded regions of the faces of the surrounding object. As their algorithm has a quadratic computational complexity in the number of edges of the model of the workspace, it requires a lot of computation time. In S. Sakane et al., the determination of the visibility region is made by performing depth buffering at each facet of a tessellated surface of a sphere centered at the target object's reference point with a pre-defined radius [10][11]. Initially, the sphere is coarsely tessellated, and then, it is tessellated finer when occlusion is detected at a facet. Facets corresponding to the occlusion avoidance are grouped into the visibility region on the spherical viewing surface. In this method, the accuracy depends heavily on the granularity of the subdivision of the sphere's surface. And, the target object is treated as a point so that there exists a difference between this result and the real visibility region.

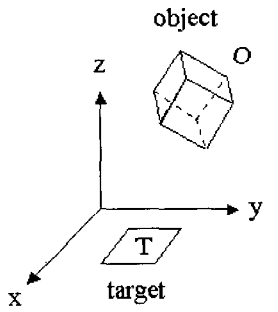
In Cartesian coordinates, we have to calculate the visibility region in x-y-z spaces. But, the calculation in 3-dimensional spaces requires much computation time. In addition, as the environment of the workspace becomes complex, the computational load grows rapidly. Therefore, for a continuous acquisition of the visual information of the target object during the task, a fast computation method is required. For the fast computation of the visibility region, we propose a spherical projection method. In the spherical projection method, the visibility region is projected onto the surface of the sphere with predefined radius r, and, is represented with two angle elements (θ and ϕ) of the spherical coordinates.

Manuscript received: Jan. 26, 2001, Accepted: May. 11, 2001.
Gil Whoan Chu: Dept. of Electrical Engineering and Computer Science, KAIST (chgh@cheonji.kaist.ac.kr)
Myung Jin Chung: Dept. of Electrical Engineering and Computer Science, KAIST (mjchung@ee.kaist.ac.kr)

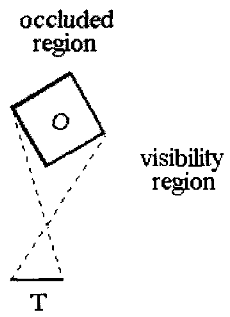
II. The visibility region

The camera should be able to 'view' the target object in order to obtain its visual information. If anything of the surrounding objects lies between the target object and the camera position, it occludes the line-of-sight of the camera. Therefore, to observe the whole of the target object, the camera should be placed within regions where no occluding object exists between the target object and the camera position. A region where the whole of the target object could be viewed without hindrance by other object is called the visibility region, while the region where the whole or some part of the target object would be occluded by the surrounding object is called the occluded region. Fig.1 shows an example of the visibility region and the occluded region in the case of that T is a target object and O is a surrounding object [12]. In Fig.1-(b), the shaded region becomes the occluded region and the remaining region becomes the visibility region.

The visibility and the occluded region are dependent on the relative positions of the target object and the surrounding objects. As the target and the surrounding objects change their position during the task, the visibility region also changes. Therefore, in order to obtain the visual information of the target object continuously during the task, the visibility region should be recalculated whenever any object changes its position and the camera modifies its configuration so as to be placed within the visibility region.



(a) model of workspace



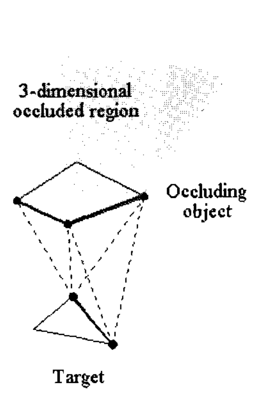
(b) visibility and occluded region

Fig. 1. Visibility region and occluded region of workspace.

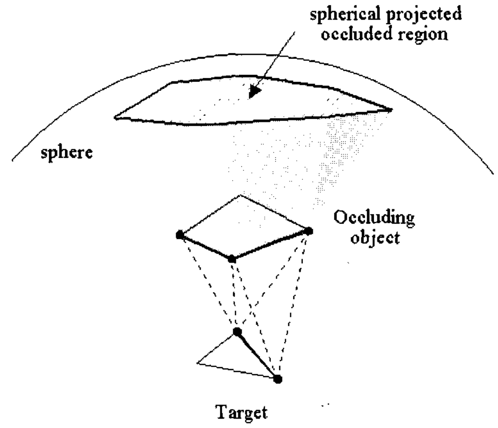
III. The spherical projection method

The visibility constraint restricts not a distance but a viewing direction of the camera position. Therefore, the camera position can be represented more simply by using the spherical coordinates instead of the Cartesian coordinates. In the spherical coordinates, the camera position and the visibility region are represented by r, θ, and ϕ. Here, the radius, r, represents the distance between the target object and the camera position, while θ and ϕ represents the viewing direction of the camera. That is to say, 3-dimensional computational spaces are divided into one dimension of the camera distance and two dimensions of the camera direction respectively.

The allowable distance between the target object and the camera position may be determined from the characteristics of the camera and the settings of the workspace. Therefore, if the inner parameter of the camera or the settings of the workspace do not change, the distance condition does not change either. In this case, the visibility region can be represented in 2-dimensional spaces (θ-ϕ spaces) instead of 3-dimensional spaces (x-y-z spaces) of the Cartesian coordinates. The reduction of the computational space from 3-dimensional to 2-dimensional conspicuously reduces the computational complexity and computation time. Fig.2 is the comparison of two methods. When finding the visibility region in the Cartesian coordinates such as Fig.2-(a), the boundary of the visibility region is represented as a set of plane equations. On the other hand, in the spherical projection method the boundary of the visibility region is represented in θ-ϕ spaces such as the slashed region in Fig.2-(b).



(a) boundary of the visibility region in 3-D spaces



(b) boundary of the visibility region in θ-ϕ spaces

Fig. 2. Comparison of the visibility region with 3-D spaces and 2-D spaces.

If the assumption is made that the workspace is modeled with convex polyhedral elements and the distance between the target object and the camera position is pre-determined as r, then the boundary of the visibility region projected onto the sphere can be obtained through two steps (Fig. 3).

■ Step 1. Find the projected vertex.

■ Step 2. Find the boundary of the projected vertices.

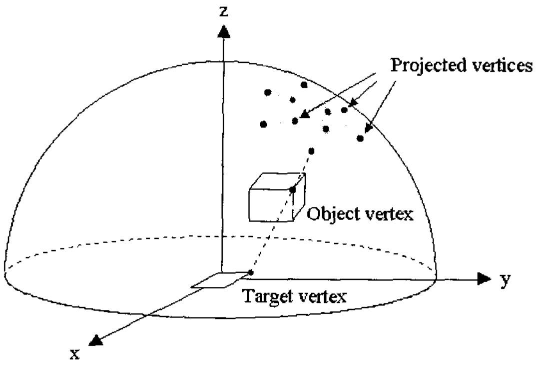
As a result of the above procedure, the set of projected vertices and its boundary is obtained in θ-ϕ spaces. Inside and outside of the boundary of the projected vertices become the occluded region and the visibility region respectively. As the position of the projected vertex and the boundary of the projected vertex have a relation to the position of the surrounding object and the radius of the sphere, there are three cases according to their position and radius r.

■ Case 1: The surrounding object lies inside of the sphere.

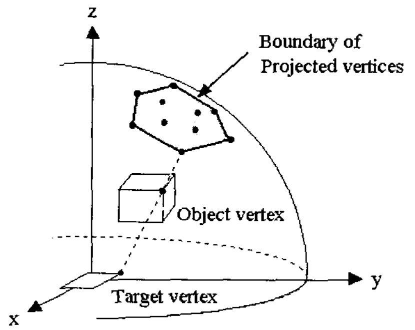
■ Case 2: The surrounding object lies across the sphere.

■ Case 3: The surrounding object lies outside of the sphere.

As the third case, if the surrounding object lies outside of the sphere it does not hinder the camera's line-of-sight nor occlude the target object. That is to say, if the object is lying farther than the radius r, it does not have an influence on the visibility region and can be ignored when computing the visi-

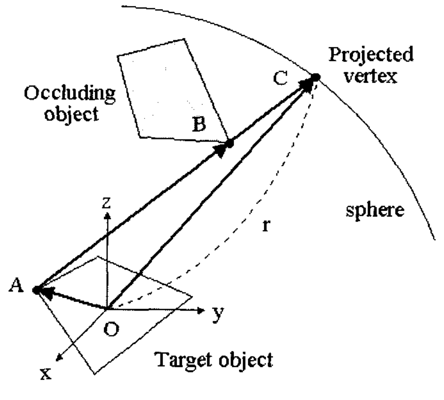


(a) find the projected vertex

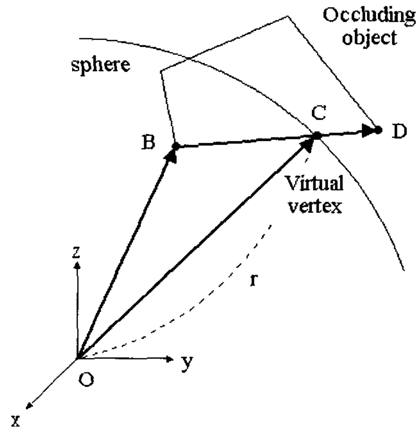


(b) find the boundary of projected vertex

Fig. 3. Spherical projection method.



(a) projected vertex of case 1



(b) virtual vertex of case 2

Fig. 4. Calculation of the projected vertex and the virtual vertex.

bility region. Therefore, when computing the boundary of the visibility region projected onto the sphere, only the first and the second case should be considered.

1. The projected vertex

If the target object and the surrounding object are modeled as a convex polyhedral, the occluded region becomes a convex polyhedral. To find the boundary of the visibility region in θ-ϕ spaces, we have to make a projection of the separating planes onto the sphere. As the separating plane is composed of the vertices of the target object and the surrounding object, the boundary projected onto the sphere is obtained by considering the projected vertex that is an intersection between the sphere and the line including one of the target object's vertices and one of the surrounding object's.

1.1 Case 1: when the object lies inside of the sphere

Fig.4-(a) shows how to find the projected vertex. Assume that A is one of the target object's vertices and B is one of the surrounding objects. Then, C becomes a projected vertex on the surface of the sphere with radius r determined by A and B. The position of A and B would be obtained from the CAD model of the workspace, and the radius of the sphere, r, would be pre-determined from the consideration of allowable distance between the camera and the target object. Then, the relation between A, B, C and r is as below.

$$\overrightarrow{OA} = (x_1, y_1, z_1) \quad (1)$$

$$\overrightarrow{AB} = (x_2 - x_1, y_2 - y_1, z_2 - z_1) = (x_2', y_2', z_2') \quad (2)$$

$$\overrightarrow{OC} = (x_3, y_3, z_3) \quad (3)$$

$$r^2 = \left|\overrightarrow{OC}\right|^2 = x_3^2 + y_3^2 + z_3^2 \quad (4)$$

The position of C could be represented using vector calculus.

$$\overrightarrow{OA} + \alpha\overrightarrow{AB} = \overrightarrow{OC} \quad (5)$$

By the substitution of Equ.1~Equ.4 into Equ.5, the proportional constant α and the position of C could be calculated.

$$\therefore \alpha = \sqrt{\frac{r^2 - (x_1^2 + y_1^2 + z_1^2)}{x_2'^2 + y_2'^2 + z_2'^2} + \frac{(x_1 x_2' + y_1 y_2' + z_1 z')^2}{(x_2'^2 + y_2'^2 + z_2'^2)^2}} - \frac{x_1 x_2' + y_1 y_2' + z_1 z_2'}{x_2'^2 + y_2'^2 + z_2'^2} \quad (6)$$

$$\therefore (x_3, y_3, z_3) = (x_1, y_1, z_1) + \alpha(x_2', y_2', z_2') \quad (7)$$

By applying the above equations to every vertex of the target object and the surrounding object, all of the projected vertices can be obtained.

1.2 Case 1: when the object lies across the sphere

If the surrounding object lies across the sphere, it can be divided into two parts: the inside and the outside of the sphere. The object lying inside of the sphere affects the visibility region while the object lying outside does not. Therefore, to find the projected vertices we generate the inside object representing the inside part of the surrounding object, and then find the projected vertices with the inside object and the target object. The inside object is composed of the vertex lying inside of the sphere and the virtual vertex which is an intersection of the surrounding object and the sphere. As Fig.4-(b), the virtual vertex appears on the edge of the object whose one vertex lies inside while another lies outside of the sphere. When B and D is the vertex lying inside and outside of the sphere respectively, the relation between B, D, and the virtual vertex C is as below.

$$\overrightarrow{OB} + \alpha \overrightarrow{BD} = \overrightarrow{OC} \quad (8)$$

The calculation of the virtual vertex, C, is similar to that of case 1. Therefore, we can find the position of C by using Equ.6 and Equ.7 in the case that A and B is replaced by B and D respectively. Note that, the proportional ratio, α, is smaller than 1 in case 2, while bigger than 1 in case 1. After the generation of the inside object, the target object and the inside object can be treated as the case 1 and the projected vertex can be found by using the same algorithm.

2. The boundary of the projected vertices

The boundary of the visibility region can be obtained by finding the boundary of the projected vertices. In spite of the convex polygonal characteristics of the projected boundary, its representation of θ-ϕ spaces does not convex any more. Therefore, to obtain the convex polygonal boundary of the projected vertices, we have to find the projected vertices that make intersections of the outmost separating planes. The determination of the outmost separating plane can be accomplished by comparing the normal vector of each plane. By repeating this process, we can obtain a set of projected vertices that makes the convex polygonal projected boundary. This algorithm is outlined below. Assume that S_{PV} is a set of every projected vertex.

■ Step 1. Set $S_c = \{\varnothing\}$.

■ Step 2. Find a projected vertex with maximum θ, $P_{\theta_{max}}$, among all of the projected vertices in S_{pv} and add it to set S_c. (Fig.5-(a))

■ Step 3. Set $P_{\theta_{max}}$ as P_2, and the vertex of the target object that makes P_2 with spherical projection as P_1.

■ Step 4. Find a projected vertex P_3 in the set S_{pv} for the plane determined by P_1, P_2, and P_3 to have the most outside normal vector and add it to the set S_c. (Fig.5-(b))

■ Step 5. Replace P_2 with P_3 and repeat step 3 until P_3 becomes $P_{\theta_{max}}$.

On all occasions, the projected vertex with maximum θ becomes an element of the projected vertices making the boundary of the visibility region. Therefore, to find the projected boundary of the visibility region, we start from this vertex in step 1 and step 2. In step 3 and step 4, the projected vertex of the outmost separating plane is determined and added in set S_c. The determination of the outmost separating plane can be accomplished by comparing the normal vector of each separating plane (Fig.5-(b)). As a result of this procedure, we obtain a set of the projected vertices that makes the boundary of the projected vertices. The inside and the outside of the boundary become the occluded region and the visibility region respectively.

IV. Simulations

In order to verify the proposed spherical projection method for finding the visibility region, several simulations were performed. By using the spherical projection method, the visibility region can be obtained quickly even though the model of the workspace is complicated or the surrounding objects change their position. In Fig. 6, the workspace is composed of a target object, surrounding objects, and two manipulators. M1 is a manipulator for doing task while M2 is an active camera system to observe the target object and workspace during the task. Each manipulator (RV-M2, Mitsubishi) is modeled with 16 solids, 388 faces, and 712 vertices. The quadrangle located at the center is the target object while the cube and the manipulator become occluding objects. In each time, the computation time of projecting the vertices onto the sphere is *36ms* with Pentium II-333MHz process. And, the determination of the boundary of the visibility region tasks about *550ms*.

In the first place, the influence of the spherical radius on the position of the surrounding object is examined. When r changes from *100mm* to *1500mm*, the alternation of the visibility region is shown in Fig. 7. As the projected vertex is represented in polar form in Fig. 7, θ of its position appears as a distance from the center of the coordinates, while ϕ appears as an angle from the horizontal axis (Fig. 7-(a)). When r=*100mm*, neither the cube nor the manipulator is included in the sphere. Therefore, all of the free space becomes a visibility region. As the radius increases the cube and the manipulator become more and more included in the sphere such that the visibility region reduces. The second simulation is to examine the visibility region when the manipulator moves during the task. According to the motion of the manipulator, the occluded region due to the manipulator also changes. Fig. 8 shows the alternation of the visibility region in accordance with the motion of the manipulator. In this case, the distance between the target object and the camera position is pre-determined as r=*1500mm*.

V. Comparison of the computational complexity

We considered the worst-case complexity of each algorithm to compare the computational complexity of the proposed spherical projection method with conventional two algorithms: the boundary-based algorithm and the decomposition-based algorithm [9]. Assume that the target object has m vertices (or edges) and the surrounding objects have altogether f faces, e edges, and v vertices, where O(f) = O(e) = O(v) = n. Then, there can be n front-facing faces, and O(mn) = O(N) locally separating planes. Bound on the worst-case computational complexity of the boundary-based algorithm and the decom-

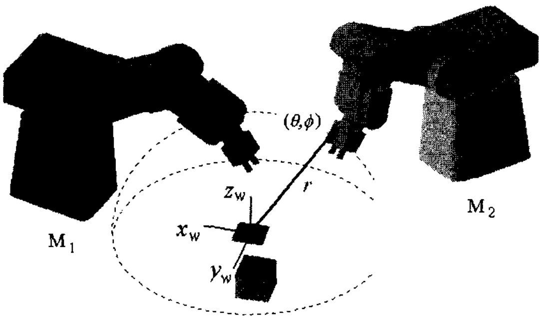


Fig. 6. Model of workspace for simulation.

position-based algorithm becomes $O(N^4)$ and $O(N^3)$ respectively. While, that of the spherical projection method is only O(N).

Until now, there has been no result on the sensor planning method applied on the on-line task because of the high computational complexity. According to the research on K. Cowan and P.D. Kovesi [9], it required about 20 seconds to calculate the camera position at each time with a time-shared VAX-8600 computer. Therefore, the sensor planning method could be applied only to a pre-defined task as an off-line method. However, as the simulation result shows the spherical projection method requires about 0.5 second. The difference of the computational complexity is due to the basic elements of the algorithm to find the boundary of the visibility region. In the boundary-based and the decomposition-based algorithms, the basic element to compute the boundary of visibility region or the occluded region is an edge or a face of the target object and the surrounding object. Therefore, in these two algorithms the surrounding objects are partitioned into faces first. And then, the boundary of the visibility region for each face is computed. Finally, by combining these boundaries and by finding their union, the complete boundary of the visibility region is determined as a set of plane equations. On the other hand, the vertex is a basic element in the proposed spherical projection method. By projecting the vertices onto the sphere and by finding the outmost boundaries of the projected vertices, the complete boundary of the visibility region is determined. To find the projected vertex is accomplished by simple vector calculus. Due to the elimination of the union procedure and the simplicity of the algorithm, the computational complexity is much lower than those of the boundary-based and the decomposition-based algorithm.

VI. Conclusions

To obtain the visual information of a target object during a task, a camera should be located and its position should be modified according to the change of the surrounding objects. The continuous observation of the target object is guaranteed in condition that the camera is located within the visibility region. For the fast computation of the visibility region and the observation of the target object with no hindrance by the surrounding objects, we proposed a spherical projection method. In the spherical projection method, the visibility region is determined by finding the projected vertex and the boundary of them. Inside and outside of the boundary become the occluded region and the visibility region respectively. Because of the vertex-based algorithm as well as the reduction of the computational space, the spherical projection method enables fast adaptation of the camera position according to the change of the surrounding objects and the continuous observation of the target object during the task. Fast computation of the visibility region enables the adaptation of the sensor planning method to an on-line task with a dynamic environment such as a tele-operating tasks.

References

[1] K. Tarabanis, R. Y. Tsai, and P. K. Allen, "Automated sensor planning for robotic vision tasks," *IEEE Proc. Int. Conf. Robotics and Automation*, pp. 76-82, 1991.

[2] R. Sharma and S. Hutchinson, "Optimizing hand/eye configuration for visual-servo systems," *IEEE Proc. Int. Conf. Robotics and Automation*, pp. 172-177, 1995.

[3] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision," *IEEE Trans. Robotics and Automation*, vol. 9, no. 1, pp. 14-35, 1993.

[4] K. Tarabanis, R. Y. Tsai, and A. Kaul, "Computing occlusion-free viewpoints," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 279-292, 1996.

[5] R. Sharma and S. Hutchinson, "On the observability of robot motion under active camera control," *IEEE Proc. Int. Conf. Robotics and Automation*, pp. 162-167, 1994.

[6] S. Yi, R. M. Haralick, and L. G. Shapiro, "Optimal sensor and light source positioning for machine vision," *Int. J. Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 122-137, 1995.

[7] Triggs and C. Laugier, "Automatic camera placement for robot vision tasks," *IEEE Proc. Int. Conf. Robotics and Automation*, pp. 1732-1737, 1995.

[8] K. Cowan, "Model-based synthesis of sensor location," *IEEE Proc. Int. Conf. Robotics and Automation*, pp. 900-905, 1988.

[9] K. Cowan and P. D. Kovesi, "Automatic sensor placement from vision task requirements," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 407-416, 1988.

[10] S. Sakane, M. Ishii, and M. Kakikura, "Occlusion avoidance of visual sensors based on a hand eye action simulator system: HEAVEN," *IEEE Proc. Int. Conf. Advanced Robotics*, vol. 2, no. 2, pp.149-165, 1987.

[11] S. Sakane, T. Sato, and M. Kakikura, "Model-based planning of visual sensors using a hand-eye action simulator system," *IEEE Proc. Int. Conf. Advanced Robotics*, 1987

[12] C. K. Cowan and A. Bergman, "Determining the camera and light source location for a visual task," *IEEE Proc. Int. Conf. Robotics and Automation*, pp. 509-514, 1989.

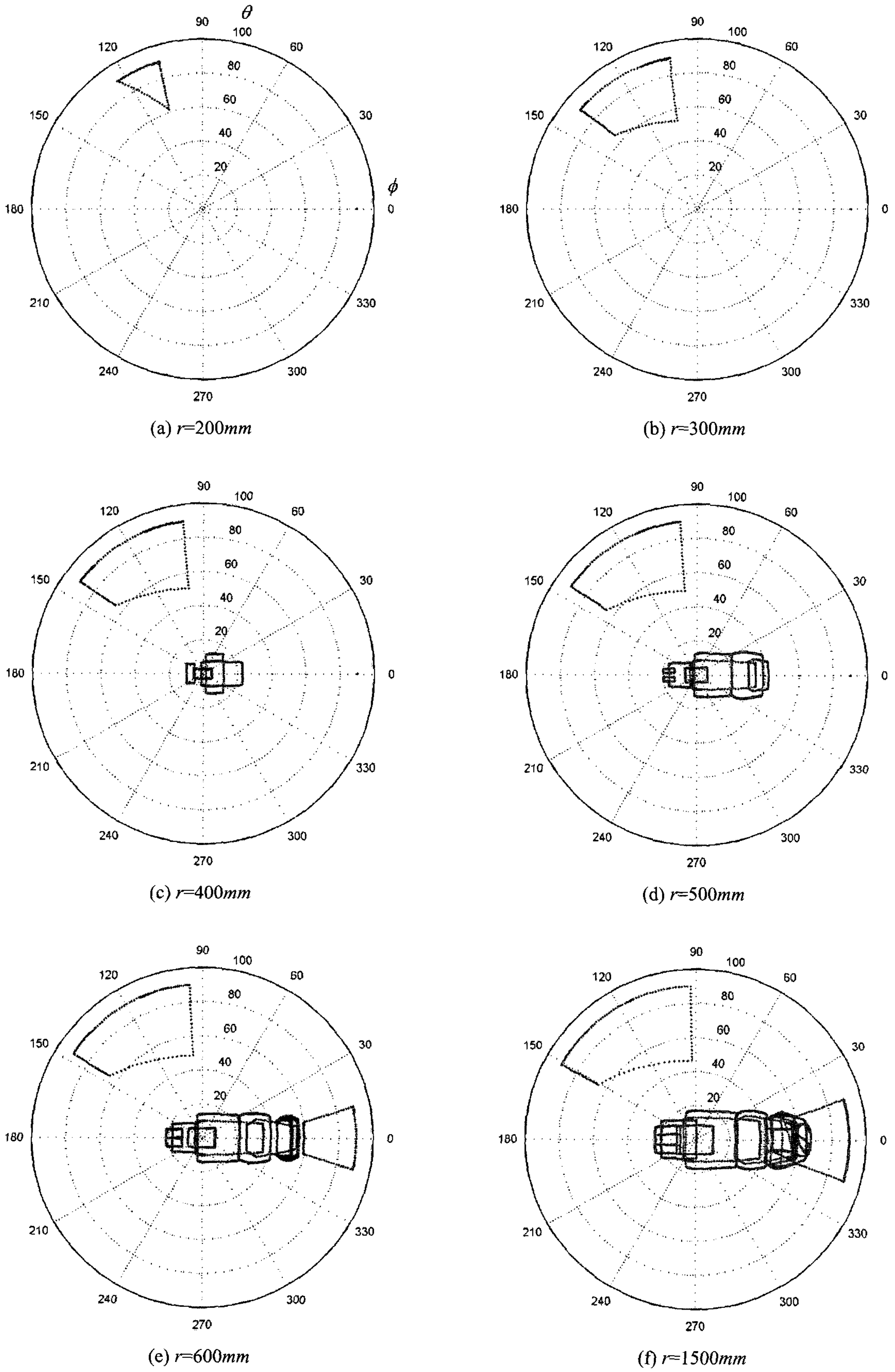
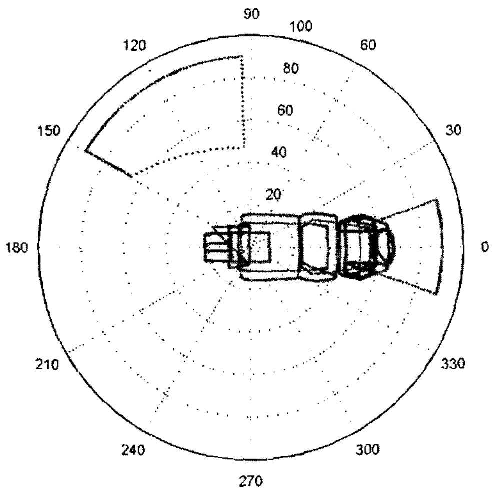
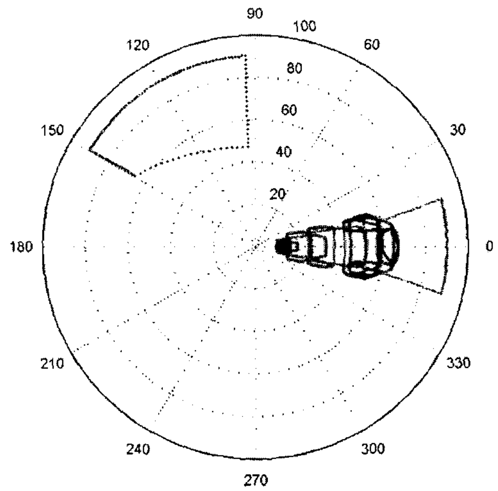


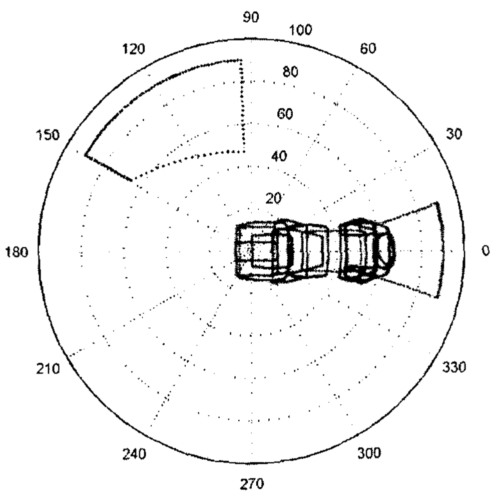
Fig. 7. The visibility region according to the change of the radius.



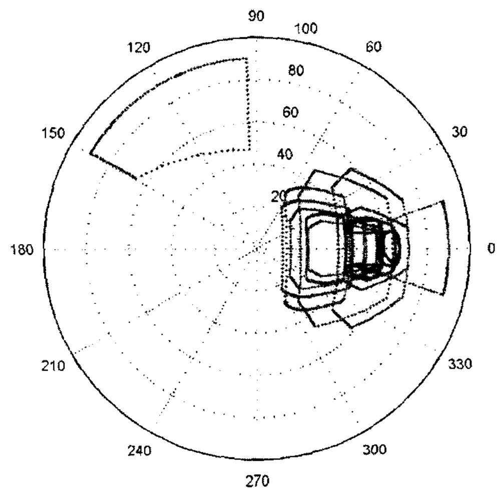
(a) motion 1



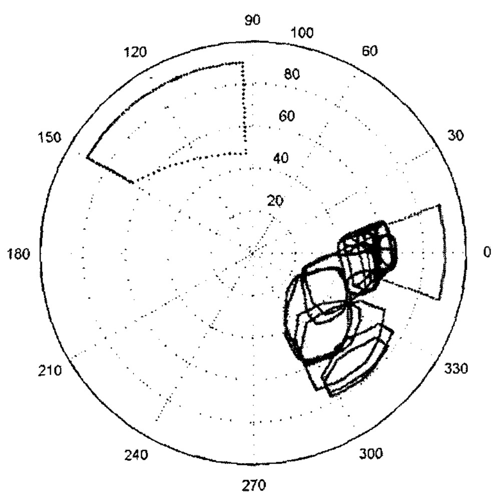
(b) motion 2



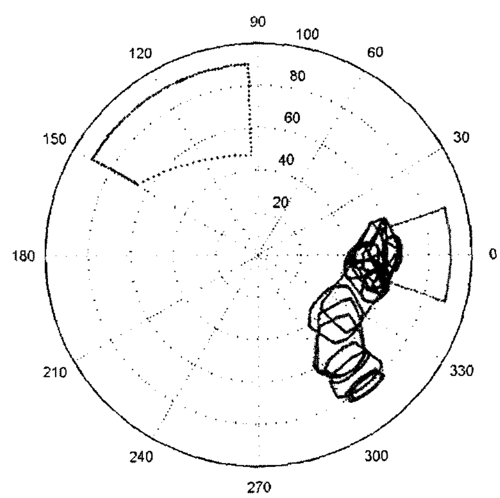
(c) motion 3



(d) motion 4

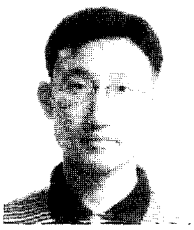


(e) motion 5



(f) motion 6

Fig. 8. The visibility region according to the motion of the manipulator with *r=1500mm*.



Gil Whoan Chu

He received the B.S. degree in electrical engineering from Pusan National University, Pusan, Korea in 1993, the M.S. and Ph.D. degrees in electrical engineering and computer science from Korea Advanced Institute of Science and Technology(KAIST), Taejon, Korea in 1995 and 2001, respectively. Since 2001, he works for the Division of Home Network Solution of Samsung Electronics Co., Ltd as a senior engineer. His research interests include sensor planning, telerobotics, and medical and rehabilitation robot.

Myung Jin Chung

He received the B.S. degree in electrical engineering from Seoul National University in 1973, the M.S. degree in electrical and computer engineering in 1977 and the Ph.D. degree in computer, information, and control engineering in 1983 both from the University of Michigan. Since 1983, he has been with the Department of Electrical Engineering and Computer Science at Korea Advanced Institute of Science and Technology, where he is currently a Professor. In 1976, he worked for the Agency for Defense Development, Seoul, as a member of technical staff and from 1981 to 1983 he worked for the Center for Robotics and Integrated Manufacturing, the University of Michigan. His research interests include intelligent control, robust control, robust path planning and control, telerobot control, and multisensor fusion.