Vis Comput (2010) 26: 629–638
DOI 10.1007/s00371-010-0489-z

ORIGINAL ARTICLE

Camera movement for chasing a subject with unknown behavior based on real-time viewpoint goodness evaluation

Maya Ozaki · Like Gobeawan · Shinya Kitaoka · Hirofumi Hamazaki · Yoshifumi Kitamura · Robert W. Lindeman

Published online: 1 May 2010

Abstract We propose a method that automatically generates a smooth chase camera movement to follow a subject, a user-controlled character or a character with unknown behaviors, in a 3D environment freely in real time. We consider three objectives in generating the smooth-camera movement: to avoid collisions with obstacles, to avoid subject occlusions, and to choose a good viewpoint for looking at the subject. We evaluate the goodness of viewpoints by using a viewpoint entropy map and choose the best viewpoint as the goal position of the camera in real time. Afterwards, we move the camera toward the goal position by following the shortest path, found by the A* algorithm, on a roadmap graph. The resulting camera movement has a high degree of freedom and fulfills the three objectives above. Our method is effective for third-person-view 3D applications in tracking the real-time movement of user-controlled characters in exploring a 3D environment.

Keywords Computer animation · 3D Virtual environment · Computer games · Camera control · Path planning · Viewpoint entropy

M. Ozaki (✉) · S. Kitaoka
Human Interface Engineering Lab., Osaka University,
2-1 Yamadaoka, 565-0871 Suita, Osaka, Japan
e-mail: maya.ozaki@gmail.com

S. Kitaoka
e-mail: skitaoka@gmail.com

L. Gobeawan
Advanced Computing, Institute of High Performance Computing,
Fusionopolis 1 Fusionopolis Way, 138632 Singapore,
Republic of Singapore
e-mail: gobeawanl@ihpc.a-star.edu.sg

H. Hamazaki
Sharp Corporation, 22-22 Nagaike-cho, 545-8522 Abeno-ku,
Osaka, Japan
e-mail: hamazaki.hirofumi@sharp.co.jp

Y. Kitamura
Research Institute of Electrical Communication, Tohoku
University, 2-1-1 Katahira, 980-8577 Aoba-ku, Sendai, Japan
e-mail: kitamura@ist.osaka-u.ac.jp

R.W. Lindeman
Computer Science, Worcester Polytechnic Institute, 100 Institute
Road, 01609-2280 Worcester, MA, USA
e-mail: gogo@wpi.edu

1 Introduction

In third-person-view 3D applications, especially 3D computer games or 3D chat systems such as Metaverse [1], a user virtually explores the 3D environment freely by controlling a character, which is the subject of the camera, while the behavior of the subject is unknown to the camera. The most common approach to having the camera capture the subject is simply to make the camera follow the subject from a fixed distance. However, in a complex environment with many static and dynamic obstacles, the camera often cannot capture the subject or the whole environment well because the camera penetrates obstacles or the obstacles block its view of the subject. To solve these problems, the camera movement could be constrained, or the camera could jump to another position. While this solves the problems of collision and occlusion, it creates a discontinuity in camera movement, causing the exploration to be distracting and difficult. In addition, the camera may not capture the subject clearly from a desirable viewpoint. Good camera work should be able to choose a good viewpoint for the camera to see the subject from, by considering the important features of the subject.

In this paper, we propose a method of generating an automatic chase camera movement based on a real-time viewpoint evaluation. We consider three main objectives in generating a smooth chase camera movement for a subject with unknown behavior:

1. To avoid collisions (or intersections) between the camera and the obstacles.
2. To avoid occlusions of the subject.
3. To move the camera to a position at which it can view the subject clearly.

In our proposed method, the goodness of viewpoints around the subject is evaluated using the concept of viewpoint entropy [15], which is a measure of the amount of important information that is visible at a viewpoint. Based on the viewpoint goodness evaluation, we choose, in real-time, the best viewpoint with the highest viewpoint entropy as the goal viewpoint, and move the camera toward that viewpoint. For the path of camera movement toward the goal viewpoint, we use a roadmap graph, which describes the camera movement space that is free of collision with obstacles. We find the shortest path on the roadmap graph, and then prevent any collisions with moving obstacles during the runtime movement of the camera. Our method produces camera movement that changes adaptively with respect to moving obstacles.

The rest of this paper is organized as follows. In Sect. 2, we discuss related works of camera control, path-finding, and viewpoint goodness evaluation. Section 3 contains the details of our integrated method for chase camera movement: viewpoint evaluation and creation of the camera movement. In Sect. 4, we present our results. Conclusions and future works follow in Sect. 5.

2 Previous work

Camera movement in 3D virtual environments has been studied in many fields. In the field of storytelling, the camera does not merely follow the subject, but also moves artistically as programmed in scripts. Many scripting languages have been developed [2, 3] for programming the camera movement in detail as desired by the story director. In order to be effective, the director has to master the given scripting language, but the learning process is time-consuming. This scripting approach does not meet our objective of producing an automatic chase camera movement in a general 3D environment.

The study of avoiding collisions with obstacles, by means of roadmap graphs, has been carried out in robotics and 3D graphics. A roadmap graph describes the collision-free space for the camera movement. Thus, collision-free camera movement can be automated by finding a path from the camera to a subject or a goal position on the roadmap graph. The roadmap graph can be created by using either a probabilistic or a deterministic method. Probabilistic methods [4, 5] find collision-free positions randomly in an environment and form the nodes accordingly on the roadmap graph. Deterministic methods [6–8] divide an environment into a set of cells and generate the roadmap graph from those cells, which are obstacle-free. Both probabilistic and deterministic methods produce collision-free camera movements, but they limit the camera movement to line segments between the nodes or cells in the roadmap graph. Consequently, the camera has a narrow movement space and low degrees of freedom in chasing a moving subject freely.

Path-finding methods that simulate a potential field of both the repulsive force from obstacles and the attractive force to the goal have also been studied [9]. By using the potential field, we can generate a smooth, collision-free path. However, the path may not lead the camera to the actual goal when the path ends at a local minimum in the potential field.

Subject occlusions, being related to the work of camera movement, pose problems such as the disappearance of the subject and the difficulty in capturing the subject fully. During important moments in a scene, subject occlusions are very undesirable. Hence, studies to solve occlusion problems have been widely carried out [10, 11]. The study reveals that occlusion-free smooth camera movement is computationally expensive. A method incorporating the concepts of the roadmap graph and the potential field to create an automatic chase camera movement [12] partially solves the subject occlusion problem by assigning an occlusion-free position as the goal position. However, the method does not solve the problem of subject occlusion when the camera is still on its way to the goal position. Moreover, the method does not consider all possible goal positions to choose a better viewpoint for the camera to look at the subject more clearly.

Choosing a good viewpoint to look at the subject is important for allowing the audience to perceive the message or information from the subject optimally. Subsequently, the evaluation of viewpoint goodness has been studied extensively in computer graphics [13, 14]. The viewpoint entropy [15], a measure of the amount of information that can be perceived from the viewpoint, is a good indicator in evaluating the goodness of viewpoints. The viewpoint entropy is determined by the number of projected surfaces and the areas of those surfaces. This approach enables an accurate evaluation of viewpoint goodness even for complicated subjects, but it is computationally expensive. So is the method using mutual information [16]. Because we are handling a moving subject in the 3D environment, it is a challenge to adopt this viewpoint goodness evaluation method within the time constraint for real-time performance.

In this paper, we propose a method for generating real-time, smooth, collision-free chase camera movements while enabling the camera to see the subject from the best viewpoint. For this purpose, we employ the concept of viewpoint entropy for real-time evaluation of viewpoint goodness to

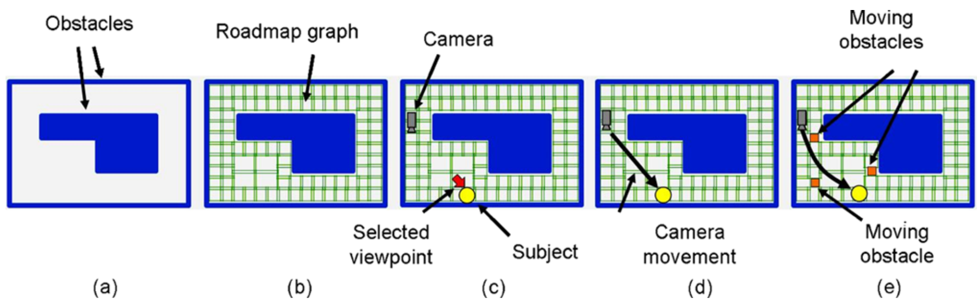


Fig. 1 Overview of our method. (**a**) detect static obstacles in the environment; (**b**) create a roadmap graph; (**c**) select a viewpoint; (**d**) generate camera movement; (**e**) adapt camera movement to dynamic environment

choose the best viewpoint, in addition to the ideas dealt with in [12].

3 Camera control

In this section, we describe our camera-control method, which generates a smooth chase camera movement without collisions or occlusions and chooses the best viewpoint for viewing the subject clearly. This method consists of three steps (see Fig. 1).

1. Create a roadmap graph.
2. Evaluate the goodness of viewpoints.
3. Generate the camera movement.

In Step 1, we create a roadmap graph by using a hierarchical cell decomposition method based on a loose tree structure. In Step 2, we evaluate the goodness of the viewpoints around the subject by using a viewpoint evaluation map in order to choose the best viewpoint as the goal position for the camera. In Step 3, based on Hooke's Law, we generate the camera movement toward the goal position.

Our method, in which users can explore freely by controlling a character (that is, a subject), supports 3D environments with the following conditions:

– The number of cameras is one.
– The number of subjects to follow at a time is one.
– There are many (an unknown number of) static and moving obstacles.
– The current positions of the subject and obstacles are known.
– Movements of the subject and the moving obstacles (positions of the next and future time instants) are unknown.
– The shape of the subject is rigid (non-deformable).
– The environment is constructed from polygons only.

These conditions enable us to focus on the movement of one camera toward one subject.

3.1 Roadmap graph generation

A roadmap graph, a set of inter-connected cells in which the camera can move freely without collisions, is generated based on the work of Hamazaki et al. [12]. We use a hierarchical cell decomposition method based on a loose tree structure, such as a kd-tree or an octree, in order to derive the roadmap graph.

3.2 Evaluation of viewpoint goodness

In order to move the camera to the best viewpoint from which important features of the subject can be viewed properly among the static and moving obstacles, we have to evaluate the viewpoint goodness in real-time. For this purpose, we use a viewpoint evaluation map to find the highest value of viewpoint goodness, which corresponds to the best viewpoint around the subject. For each viewpoint, the viewpoint goodness is evaluated by calculating the viewpoint entropy [15]. To do so, for each viewpoint around the subject, the surface of the subject is projected to a 2D view plane as seen from that viewpoint, and then the viewpoint entropy E is calculated as follows:

$$E = -\sum_{i=0}^{N} \frac{A_i}{S} \log \frac{A_i}{S},$$

where N is the number of facets of the subject, A_i is the projected area of facet i, and S is the area of the 2D view plane (screen).

The cost of this calculation for all viewpoints is very high. Furthermore, in real time, moving obstacles may often obstruct the view of the subject and keep changing the goodness of viewpoints around the subject. Repeating the viewpoint evaluation for each time instant in the runtime is too time-consuming for real-time performance. Hence, an efficient algorithm for speeding up the calculation of viewpoint evaluation is necessary. The following is the process to be followed in order:

1. (Offline) Calculate the viewpoint evaluation values for each viewpoint around the subject without the presence of any obstacles in the environment.
2. (Runtime) Obtain the positions of all obstacles relative to the subject.
3. (Runtime) Modify the evaluation values obtained in step 1 based on the information obtained in Step 2.

In Step 1, we isolate the subject from the environment and the obstacles. We also define an arbitrary number of

Fig. 2 Viewpoint entropy maps and occlusion map

viewpoints, enclosing the subject, with different distances (radii) and directions from the subject. In our method, we define four radii for each viewing direction around the subject. Subsequently, we calculate the viewpoint entropy for each viewpoint around the subject, and normalize the entropy values with respect to the largest entropy value. The higher the entropy value, the better the viewpoints for capturing important features of the subject, and vice versa.

The viewpoint entropy values of four viewpoints, at the four radii along one direction, are scaled to BGRA color component values, which are aggregately mapped to one colored pixel of a texture map. Overall, the viewpoint entropy values of all viewpoints are mapped as color values to the texture map, by using paraboloid mapping [17]. We refer to this texture map as the unoccluded viewpoint entropy (UVE) map (for example, Fig. 2(a)). The composition of the UVE map is illustrated in Fig. 3. The resolution of the UVE map depends on the number of evaluated viewpoints.

We assume that the subject does not deform over time, so without occlusions, the viewpoint entropy around the subject remains the same when it is isolated from the dynamic environment. Therefore, the UVE map will not change during runtime and Step 1 can be performed offline.

Steps 2 and 3 are carried out during runtime. In Step 2, we use paraboloid mapping again to generate a grayscale subject-centered environment map, referred to as an occlusion map (see Fig. 2(b)), which stores the normalized distances of all obstacles from the subject. The location of each pixel in the occlusion map corresponds to a viewing direction to the subject in the UVE map, while the value of each pixel ranges from 0 (black) to 1 (white). The value 0 corresponds to the zero distance of an obstacle from the subject, while the value 1 indicates the largest distance (in the UVE map) and beyond, of an obstacle from the subject.

In Step 3, the UVE map and the occlusion map are multiplied to produce the occluded viewpoint entropy (OVE) map (Fig. 2(c)). On the OVE map, each pixel is derived from the BGRA color components of the corresponding pixel in the UVE map, which are scaled according to the occlusion information on the occlusion map. In this way, we can obtain the effective viewpoint entropy values for all viewpoints in real time. We can then move the camera to the best viewpoint with the highest viewpoint entropy values on the OVE map, in order to avoid subject occlusion and to capture the subject clearly.

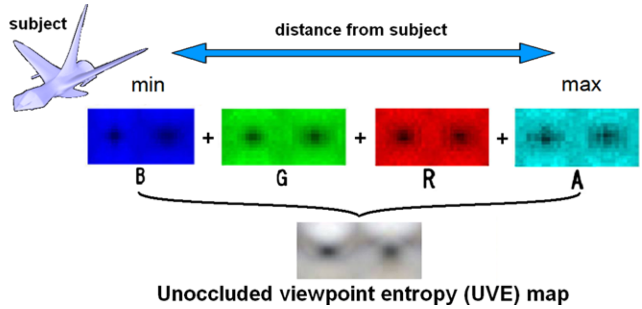


Fig. 3 Composition of UVE map

3.3 Generating camera movement

In this section, we describe a method for generating a camera path to a goal position based on the roadmap graph as proposed by Hamazaki et al. [12]. The procedure is described below:

1. From the roadmap graph, extract the start cell that overlays the camera and the end cell that overlays the best viewpoint position around the subject.
2. Find the shortest path between the start and end cells using the A* algorithm.
3. Move the camera inside the path cells toward the end cell based on the total affecting force derived from a potential function and a penalty function.

In Step 1, we detect the intersection between the bounding box of the camera and a cell in the roadmap graph to be the starting cell for the path-finding. We also extract the end cell that contains the position of the best viewpoint around the subject.

In Step 2, we use a path-finding algorithm to create the shortest path of cells in the roadmap graph from the starting cell to the ending cell. We use the A* algorithm for the path-finding because it is able to produce a path with minimum cost based on user-assigned cost functions on the nodes in the search graph.

The camera path found by the A* algorithm is not composed of line segments but cells, so we have to create a trajectory of camera movement in the cell space. In Step 3, we create a trajectory of camera movement that conforms to Hooke's Law, by using a potential function and a penalty function. The potential function determines the force that draws the camera toward the current best viewpoint position. The penalty function determines the force that constrains the camera inside the cells along the path toward the end cell.

We adapt the generated camera path to the dynamic environment with moving obstacles by the following modification:

1. Detect the cells in the roadmap graph that intersect with the bounding boxes of moving obstacles;

(a) Sponza Atrium (b) House architecture

Fig. 4 Environments

2. If the camera is inside those cells, generate a repulsive force along the surface normal of a sphere of a certain radius surrounding each moving obstacle.

4 Results and discussion

In this section, we demonstrate and analyze the camera movement produced by our proposed camera work in static and dynamic environments. First, we compare two roadmap graphs created by two hierarchical cell decomposition methods: one based on a loose octree and another based on a loose kd-tree. The two roadmap graphs are compared in terms of the resulting camera movement space. Next, we analyze the effectiveness of our camerawork in producing a collision-free camera path and choosing a good viewpoint to view the subject clearly.

The 3D environments used in our experiment are shown in Fig. 4(a)[1] and 4(b)[2], the subjects are shown in Fig. 5, and the 3D environments with moving obstacles are shown in Fig. 6. We define 32×16 viewpoints (four viewpoints at four distances per viewing direction) around each subject for the viewpoint goodness evaluation; hence, it is the same for the resolution of the UVE map, the occlusion map, and the OVE map. Our PC specifications are as follows: 2.13 GHz CPU, 2 GB RAM, and 256 MB NVIDIA GeForce 7900 GS graphics card. Our camerawork program with the specified setting runs on this system at an average frame rate of 56 fps. When the viewpoint goodness evaluation feature is turned off, the frame rate is similar, at 60 fps.

4.1 Results of roadmap graph creation

To create a roadmap graph, we have the choice of using an octree or kd-tree to subdivide the 3D environment into a set of cells. We analyze both tree methods for their effectiveness in creating the cells by comparing the sizes and the number of cells at a level of depth, as well as the space-filling ratio of the cells.

[1] http://hdri.cgtechniques.com/~sponza/files/.

[2] http://lava.ds.arch.tue.nl/lava/.

(a) Ghost model (b) Spaceship model

Fig. 5 Subjects

Fig. 6 Dynamic virtual environment includes moving obstacles

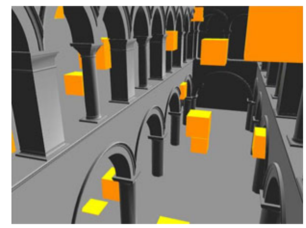
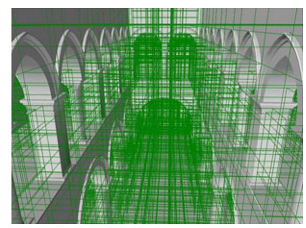


Fig. 7 Created roadmap graph (*green lines*) in the Sponza Atrium



First, we analyze the effectiveness of both trees for static environments. In order to compare the two tree methods in generating roadmap graphs with the same number of unit cells, we use the octree of a certain depth and the kd-tree of three times the depth of the octree. Table 1 shows the results of the roadmap graph creation for two environments: the Sponza Atrium (Fig. 4(a)) and the house architecture, using both the kd-tree and octree methods. Figure 7 shows the created cell-set in the Sponza Atrium. We observe that there is little difference in the space-filling ratios for both methods, and sufficient space (that is, relatively high space-filling ratio value) for the camera to move in the 3D environment, especially in the case of high maximum depth. However, in the case of low maximum depth, the camera movement space in the roadmap graph tends to decrease because the cells do not pack the obstacles tightly. We deal with this problem by increasing the maximum depth. However, high maximum depth translates into more cells generated for the roadmap graph, and the subsequently higher cost of path-finding. From Table 1, it seems that for approximately the same space-filling ratios, the kd-tree produces fewer cells compared to the octree in the static environment; hence, it appears that the kd-tree is faster at path-finding. At the same time, the cells in the kd-tree are larger in size compared to those in the octree; hence, the kd-tree reduces the camera movement space more than the octree does. Therefore, both

Table 1 Result of creating a set of cells

Environment	*k*d-tree			octree		
	depth	number of cells	space-filling ratio	depth	number of cells	space-filling ratio
Sponza Atrium	12	532	0.310	4	774	0.346
	15	3806	0.499	5	4546	0.461
	18	30594	0.641	6	62980	0.684
House architecture	12	628	0.424	4	1626	0.488
	15	5458	0.726	5	9517	0.728
	18	19081	0.826	6	40517	0.847

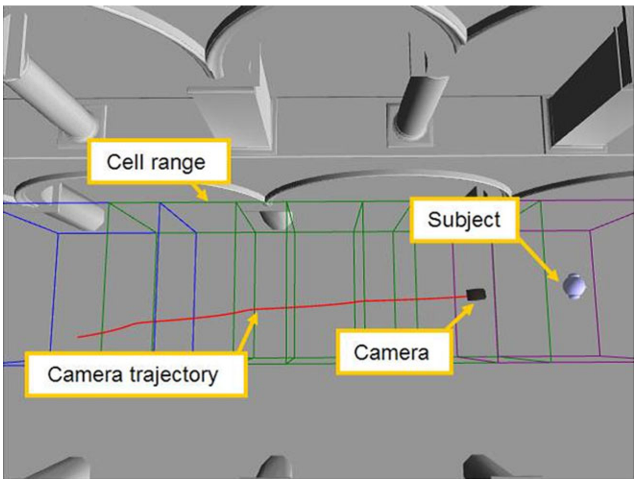


Fig. 8 View of Sponza Atrium from above with camera trajectory (*red line*) and cells (*blue* is starting cell, *purple* is ending cell, and *green* are shortest path cells)

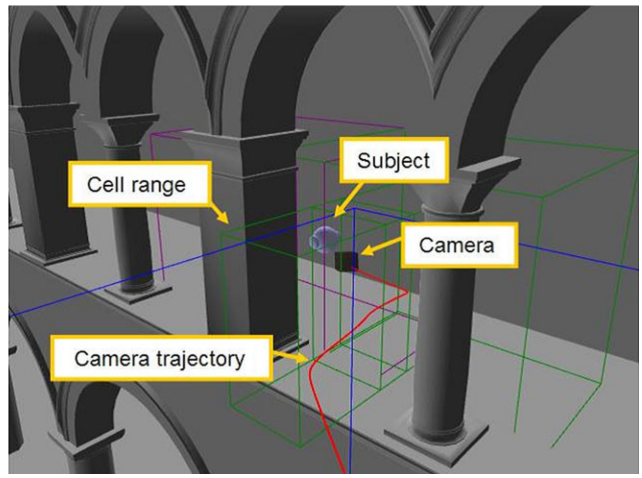


Fig. 9 Camera trajectory near obstacles

the *k*d-tree and octree are suitable for different dynamic environments.

4.2 Results of collision-free camera path

Next, we analyze the effectiveness of the camera path produced by our camerawork in avoiding collisions with obstacles. Figure 8 shows an example of the generated camera movement in the Sponza Atrium without the presence of moving obstacles. We use the Euclidean distance between the subject and the cells of the roadmap graph as a cost function of the A* algorithm in finding the shortest path from the starting cell to the ending cell.

Figure 9 shows a static environment with a smooth, collision-free camera trajectory in the space near the static obstacles, while Figs. 10 and 11 show the trajectory of the camera in a dynamic environment with moving obstacles. From this result, we confirm that our camerawork achieves the shortest camera path in following the subject while avoiding collisions with obstacles in the dynamic environment.

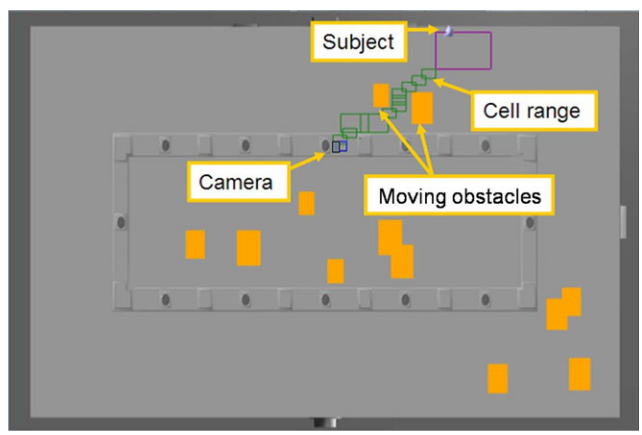


Fig. 10 Top-down view of first floor and part of second floor. We confirm that the path did not intersect with any moving obstacles

4.3 Results of viewpoint evaluation

We analyze the effectiveness of our camerawork in using the viewpoint goodness evaluation to choose the best viewpoint. The values of the viewpoint goodness evaluation are stored in a UVE map, as shown in Fig. 12(a). At runtime, the occlusion map (Fig. 12(b)) for the subject is combined with the UVE map to produce an OVE map (Fig. 12(c)) in the dynamic environment (Fig. 12(d)). Figure 13 shows some snapshots of an airplane subject from several viewpoints, together with their viewpoint goodness evaluation values. In this case, we select the viewpoint with the highest evaluation values as the goal position for the camera in real-time.

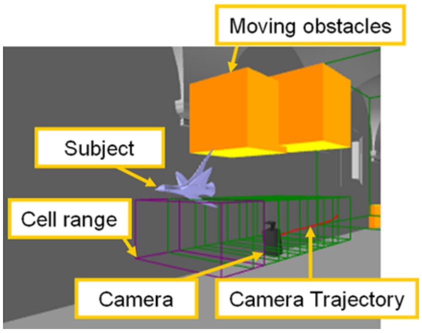
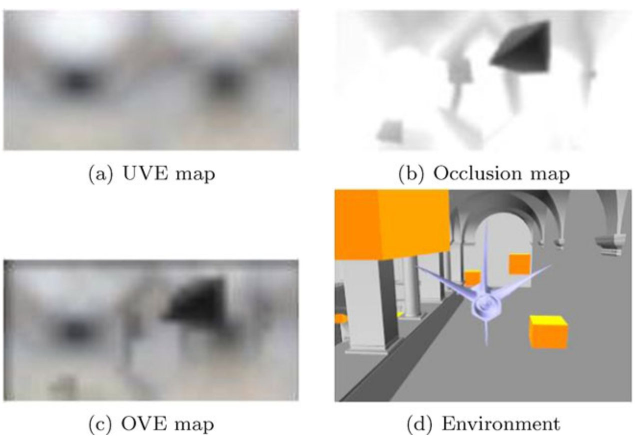


Fig. 11 Camera trajectory in dynamic environment



(a) UVE map

(b) Occlusion map

(c) OVE map

(d) Environment

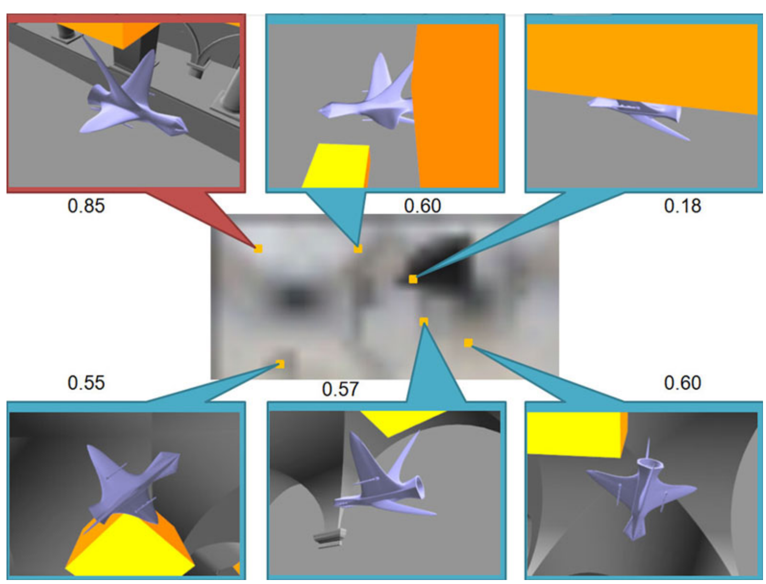
Fig. 12 Generated maps and environment

Finally, we compare the camera views in two scenarios: camerawork with and without viewpoint goodness evaluation. Figure 14 compares the viewpoint goodness evaluation values for the two scenarios. In this figure, we observe that throughout a series of continuous camera movements in the space, most of the time, the camerawork with viewpoint goodness evaluation places the camera at better viewpoints with relatively high viewpoint goodness evaluation values (hence the clear camera view of the subject), compared to the camerawork without viewpoint goodness evaluation. Figure 15 shows the corresponding camera views at some viewpoints for the two scenarios. In this figure, we observe that in the camerawork with a viewpoint goodness evaluation scenario, the camera moves to avoid occlusion of the subject, while that without a viewpoint goodness evaluation scenario does not try to avoid occlusion, as seen in Frames 12 and 18. The comparison of the two scenarios shows that our camerawork captures the subject clearly and avoids long occlusion of the subject. Occasionally, some occlusion is unavoidable, especially when the camera, without planning, has to move away to avoid a collision with an approaching obstacle. This explains the occasional drops in viewpoint goodness evaluation values at certain frames as seen in Fig. 14.

5 Conclusions and future work

In this paper, we presented a method for generating smooth movement of a camera that follows and views a subject with unknown behaviors from a good viewpoint while avoiding collisions with obstacles. First, we created a collision-free roadmap graph in the form of a set of cells using a loose-tree-based method. Next, we evaluated the goodness

Fig. 13 Camera view at several viewpoints with their viewpoint goodness evaluation values on the OVE map (Fig. 12(c))



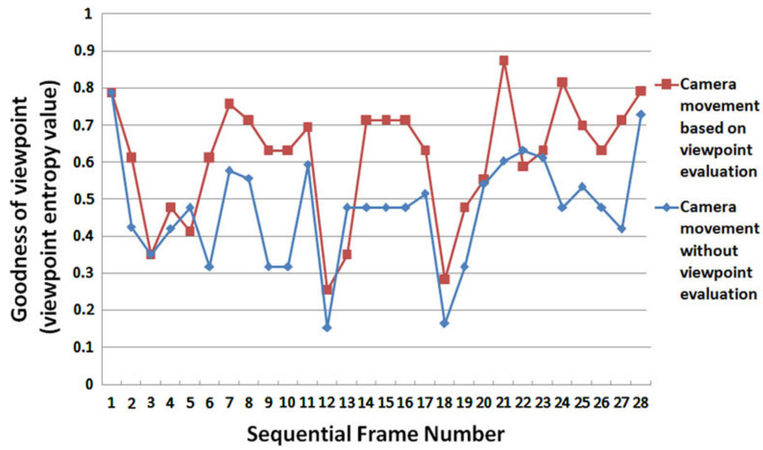
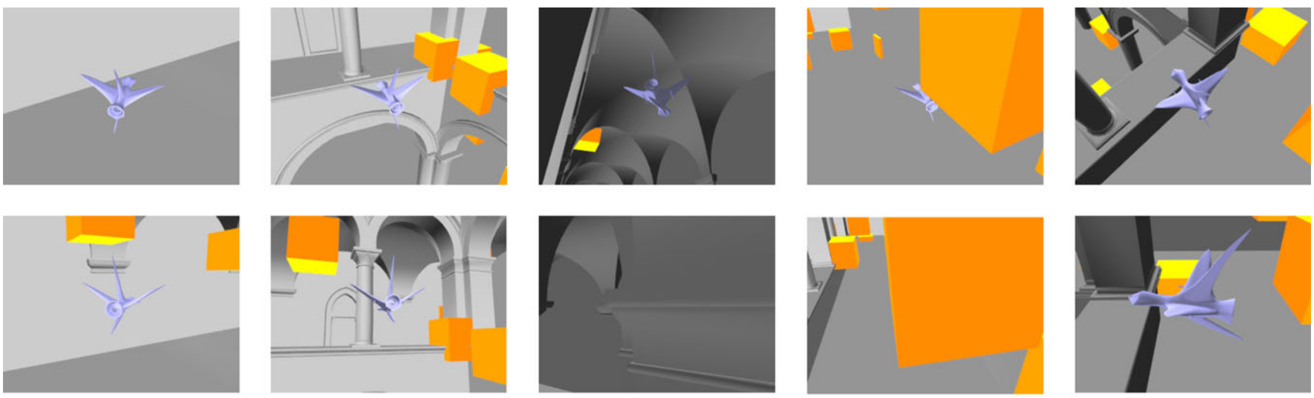


Fig. 14 Comparison between camera movement based on viewpoint goodness evaluation and that without viewpoint goodness evaluation, in terms of viewpoint entropy values as the camera follows a subject throughout a series of display frames



(a) Frame 2 (b) Frame 10 (c) Frame 12 (d) Frame 18 (e) Frame 21

Fig. 15 The images on the *upper row* show the camera views when viewpoint goodness evaluation is performed, and the *bottom* ones show the camera views when the evaluation is not performed. In the *upper row* images of columns (**a**), (**b**), and (**e**), the camera captures the subject from the top because the original viewpoint evaluation value of the subject is the highest at the top. Also, in the *upper row* images of columns (**c**) and (**d**), the camera moves to a better viewpoint at which it can still view the subject when occlusions occur, compared to the camera views shown in their corresponding lower row images

of viewpoints around the subject to choose the best viewpoint. Afterwards, we found the shortest path of cells from the camera to the best viewpoint. By controlling the camera inside the cells along the shortest path by using a potential function and a penalty function, we generated smooth chase camera movement. The movement was then modified when the camera encountered moving obstacles in the dynamic environment.

As a result, our camerawork produced smooth, collision-free camera movement and a better subject view for the camera to see important features of the subject in real time. In our method, the generation of a roadmap graph and the evaluation of viewpoint goodness operated effectively to achieve our objectives of avoiding collisions with obstacles, avoiding subject occlusion, and viewing the subject from a good viewpoint clearly. Our method can be applied to 3D games, in which a user explores a 3D environment by controlling a character.

In future work, we would like to further expand our method to generating cinematographic camera movements such as camera panning motion to rotate and cut away to another scene. We believe that in this way, the proper camera movement will make the user's experience more realistic and artistic. Furthermore, we would like to use more parameters, such as lighting, textures, and silhouettes, to evaluate the viewpoint goodness more completely. We will also extend our camerawork to handling deformable subjects. Finally, we would like to predict the movement of the subject to choose good viewpoints more quickly.

References

1. Sonvilla-Weiss, S.: (IN)VISIBLE: Learning to Act in the Metaverse. Springer, Berlin (2008)
2. Christianson, D.B., Anderson, S.E., He, L.W., Salesin, D.H., Weld, D.S., Cohen, M.F.: Declarative camera control for auto-

matic cinematography. In: Proceedings of the National Conference on Artificial Intelligence, pp. 148–155 (1996)
3. Amerson, D., Kime, S.: Real-time cinematic camera control for interactive narratives. In: The Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment (2001)
4. Kuffner, J.J. Jr., LaValle, S.M.: RRT-connect: an efficient approach to single-query path planning. IEEE International Conference on Robotics and Automation, vol. 2 (2000)
5. LaValle, S.M., Branicky, M.S., Lindemann, S.R.: On the relationship between classical grid search and probabilistic roadmaps. Int. J. Robot. Res. **23**, 673 (2004)
6. Warren, C.W.: Fast path planning using modified A* method. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 662–667 (1993)
7. Andujar, C., Vazquez, P., Fairen, M.: Way-finder: guided tours through complex walkthrough models. Comput. Graph. Forum **23**, 499–508 (2004)
8. Rosell, J., Vazquez, C., Perez, A.: C-space decomposition using deterministic sampling and distance. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 15–20 (2007)
9. Kitamura, Y., Tanaka, T., Kishino, F., Yachida, M.: 3-D Path planning in a dynamic environment using an octree and an artificial potential field. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, pp. 474–481 (1995)
10. Christie, M., Norm, J.: A semantic space partitioning approach to virtual camera control. In: Proceedings of the Annual Eurographics Conference, pp. 247–256 (2005)
11. Marchand, E., Courty, N.: Controlling a camera in a virtual environment. Vis. Comput. **18**, 1–19 (2002)
12. Hamazaki, H., Kitaoka, S., Ozaki, M., Kitamura, Y., Lindeman, R.W., Kishino, F.: Extracting camera-control requirements and camera movement generation in a 3D virtual environment. In: Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology, pp. 126–129 (2008) (short paper)
13. Barral, P., Dorme, G., Plemenos, D.: Scene understanding techniques using a virtual camera. In: Proceedings of the Eurographics' 00, vol. 2, p. 4 (2000) (short presentation)
14. Roberts, D.R., Marshall, A.D.: Viewpoint selection for complete surface coverage of three dimensional objects. In: Proceedings of the British Machine Vision Conference, pp. 740–750 (1998)
15. Vázquez, P.P., Feixas, M., Sbert, M., Heidrich, W.: Viewpoint selection using viewpoint entropy. In: Proceedings of the Vision Modeling and Visualization Conference, pp. 273–280 (2001)
16. Feixas, M., Sbert, M., González, F.: A unified information-theoretic framework for viewpoint selection and mesh saliency. ACM Trans. Appl. Percept. **6**(1), 1–23 (2009)
17. Heidrich, W., Seidel, H.P.: View-independent environment maps. In: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware (1998)

Maya Ozaki obtained both her B.Sc. in Engineering and M.Sc. degrees in Informatics from Osaka University, Japan, in 2008 and 2010, respectively. Since April 2010, she has been an engineer in the area of chemical industry at Sumitomo Chemical, Japan. Her main interests are in the fields of computer graphics, especially 3D virtual environment and camera work.

Like Gobeawan obtained both her B.Comp.Eng. and Ph.D. in Computer Engineering from Nanyang Technological University, Singapore, in 2004 and 2010, respectively. She is currently a research engineer in the area of digital modelling and visualization at Institute of High Performance Computing, Singapore. Her research interests include computer graphics, especially real time rendering and large scale visualization, computational geometry, and geometry processing.

Shinya Kitaoka received M.Sc. and Ph.D. in Informatics from Osaka University, Japan, in 2007 and 2010, respectively. Since April 2010, he has been an engineer in high performance computing and computer graphics at Prometech Software, Inc. His research interests include computer graphics, especially global illumination, camera work, and human figure control.

Hirofumi Hamazaki obtained both his B.Sc. in Engineering and M.Sc. in Informatics from Osaka University, Japan, in 2007 and 2009, respectively. In 2009, he joined Sharp Corporation, Japan. He has been working on equipment cooperation. His research interests are in the fields of computer graphics, especially 3D virtual environment and camera work.

Yoshifumi Kitamura received M.Sc. and Ph.D. degrees in Engineering from Osaka University in 1987 and 1996, respectively. From 1987 to 1992, he was at the Information Systems Research Center of Canon Inc. From 1992 to 1996, he was a researcher at the ATR Communication Systems Research Laboratories. From 1997, he was an Associate Professor at the Graduate School of Engineering and Graduate School of Information Science and Technology, Osaka University. Since April 2010, he has been a Professor at the Research Institute of Electrical Communication, Tohoku University, Japan. His current research interests include interactive content design and 3D user interfaces.

Robert W. Lindeman received his B.A. in Computer Science from the Michtom School of Computer Science in 1987. He earned an M.S. in Systems Management, from the Institute of Safety and Systems Management, The University of Southern California in 1992. He finished his Sc.D. in Computer Science in May 1999, in the Department of Computer Science at The George Washington University, and worked in the same department as an assistant professor until June 2005. He joined the Department of Computer Science at Worcester Polytechnic Institute in July 2005. His area of research is interaction techniques for immersive virtual environments.