Journal for Geometry and Graphics
Volume 11 (2007), No. 1, 45–57.

Development of a High Quality Expression System of Natural Elements Using the Real-Time Simulation Method

Tomohiro Fukuda[1], **Kazuhiro Sakata**[2], **Wookhyun Yeo**[3], **Atsuko Kaga**[1]

[1] *Graduate School of Engineering, Osaka University*
2-1, Yamadaoka, Suita, Osaka, 565-0871, Japan
email: fukuda@see.eng.osaka-u.ac.jp

[2] *ORIX Corporation*
2-4-1, Hamamatsu-cho, Minato-ku, Tokyo, 105-6135, Japan
email: kazuhiro_sakata@ore.orix.co.jp

[3] *Dept. Housing and Interior Design, Yonsei University*
134, Shinchon-dong, Seodaemun-gu, Seoul, 120-749, Korea
email: kazuhiro_sakata@ore.orix.co.jp

Abstract. In this paper for a real-time simulation by VR (Virtual Reality) technology an expression method of sunlight, shadows, grass, and water is developed. These are elements requiring especially high operating frequency in a landscape simulation. For shadow expression, two functions are required: the movement of a light source to duplicate a solar orbit, and the rendering of shadows. The movement of a light source draws an orbital equation from a solar position function, and moves a light source using this equation. The rendering of a shadow is drawn by the shadow volume method. This technique extends the vertex outline on a 3D model from a light source, and creates a shadow volume. This shadow volume is rendered to a stencil buffer. For grass expression, texture mapping is performed by a shader program using HLSL (High Level Shader Language), raising the transparency of each polygon. As it goes to the leaf point it simultaneously creates 20 polygons in piles in the normal direction. Furthermore, texture mapping is performed, raising the transparency of each polygon as it goes to the leaf point. By this technique, plural thin polygons can express a thick object. For water expression, refraction of light is expressed with HLSL by carrying out bump mapping of the image under the water surface to the material of a water surface object. Reflection of light is realized by piling up the upside-down picture

ISSN 1433-8157/$ 2.50

which sets a dummy camera to the position which reversed the camera to the water surface, and a dummy camera draws on it by HLSL to the material of a water surface object. An expression system of shadow, grass and water surfaces could be developed, and high quality expression that can be conventionally used for design review of a close-range view was realized.

Key Words: Environmental design, real-time simulation, virtual reality, expression of natural elements, close-range view, system development

MSC: 68U05, 51N05

1. Introduction

In recent years, social concern about the landscape has been increasing. The landscape law, which is a law to protect the landscape, was enacted in Japan in 2004. While the concern about landscape increases, the necessity for integrated environmental design incorporating architecture, engineering works, city space, landscape, etc. is increasing. With the rise in concern for the landscape, the use of real-time simulation methods employing VR (Virtual Reality) technology has spread as a method of evaluating a landscape [1]. The real-time simulation method can enable a design proposal to be understood intuitively through realistic expression similar to 3DCG (3-dimensional Computer Graphics). In addition, it is interactive. Viewpoints can be changed and studied through interactive operation, and the validity of the system as a landscape evaluation tool is now accepted [2]. On the other hand, an inferior point to 3DCG, is that since a high-speed rendering of more than 10 fps (frames per second) is needed, at present, the quality of 3DCG has not been reached for expression of natural elements of the close-range view that require detailed rendering. In order to realize detailed rendering in a real-time simulation, hardware and software which enable higher speed rendering processing are required. In recent years, in addition to the CPU (Central Processing Unit), progress in development of the GPU (Graphics Processing Unit) has been remarkable, and throughput and speed became quick twice every 6 months. Furthermore, the use of a programmable shader was attained in DirectX9 [3]. Improvement has been advanced while Virtools™ Dev, which the authors use as a development toolkit for real-time simulation, also corresponds to them [4]. This paper takes account of the above background in real-time simulations used for landscape evaluation, and after clarifying the problem of natural element expression in a close-range view, develops close-range view expression methods of shadow, grass, and water in order to solve the problems involved. Furthermore, the developed expression methods are applied to environmental design contents, and the influence on fps is considered. Finally, user evaluation by a questionnaire is performed (Fig. 1).

2. Analysis of expression of the real-time simulation method

Composition elements treated by an environmental design are divided into *artificial elements* and *natural elements*. As for artificial elements, such as buildings, bridges, and roads, design information, such as size and form, is correctly defined on the drawing. Therefore, correct modeling is possible. Moreover, since a material is made artificially, it has comparatively simple textures and color. On the other hand, the form of a natural element is not regular, and color also has complicated features. For these reasons, it is supposed that a realistic expression of a natural element is difficult compared with an artificial element. Moreover,

Figure 1: Design review scene with real-time simulation system. The left photo shows the note PC type and the right photo shows the large dome screen type

although detailed expression is not needed in a middle-range view or a distant view, expression with accurate detail is called for in a close-range view. However, the required reality may not be achievable in the present condition. An expression example of a real-time simulation is shown in Fig. 2. Moreover, examples of inadequate expression are shown in Table 1. In this paper, an expression method of sunlight shadow, grass, and water is developed. These are elements requiring an especially high operating frequency in a landscape simulation.

Element	*Classification*	*Contents*
soil	close-range view	unevenness of surface of the earth; granular expression of sand or soil
water	close-range view	expression of the water surface including reflection or refraction
plant	close-range view	expression of the textures of plants with fine composition, such as grass and flowers
climate	close-range, middle-range, distant view	expression of precipitation, such as snow and rain; expression of falling snow and rain
sunlight	close-range, middle-range view	a plant shadow; expression of the sunlight shadow accompanied by change

Table 1: Items of expressional inadequacy

3. System development of natural element expression

3.1. Shadow

First, the movement of light source data is described. Considering the position of the appearance of the sun seen from a watcher on the earth, when the observation point on the earth

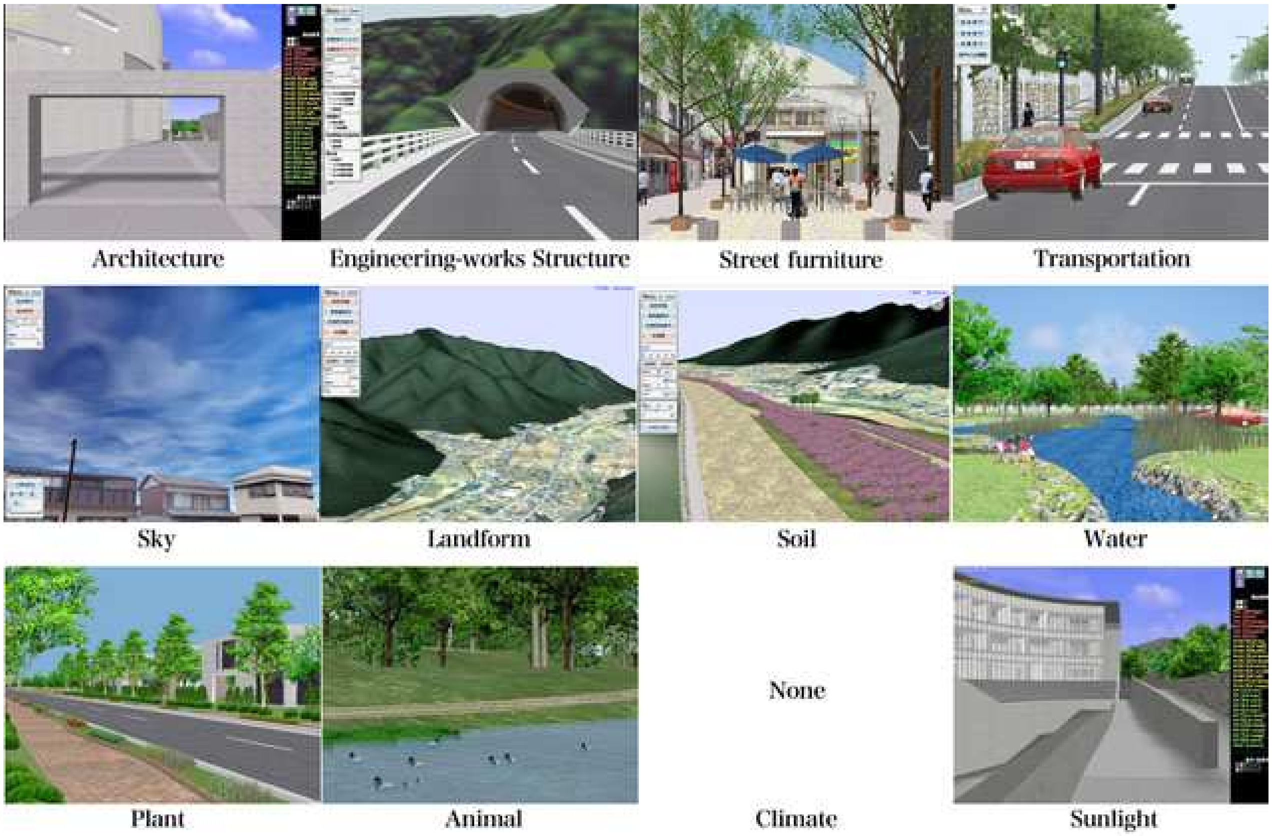

Figure 2: Expression examples in a real-time simulation

is fixed, the sun will move across a hemisphere called a celestial sphere with a virtual radius. The solar orbit changes every season with the inclination of the earth's axis. For example, the latitude of Japan and near north latitude 23.5 degrees, as shown in Fig. 3, it changes. The position of the sun on this celestial sphere is determined by two parameters, the *solar altitude* (h) and the *solar direction* (A), as shown in Fig. 4. Here, the solar altitude (h) is the height from the horizon, and is always positive. The solar direction (A) for true south makes 0°, and the west side is set as positive and the east as negative. Generally among three elements such as the latitude of the observation point, the solar celestial declination on the observation day, and hour angle of observation time, the following two formulas are materialized.

$$\sin h = \sin\varphi * \sin\delta + \cos\varphi * \cos\delta * \cos t \quad (1)$$

$$\sin A = \sin t * \cos\delta / \sin h \quad (2)$$

namely,

$$h = \arcsin\left(\sin\varphi * \sin\delta + \cos\varphi * \cos\delta * \cos t\right) \quad (3)$$

$$A = \arcsin\left(\sin t * \cos\delta / \sin h\right) \quad (4)$$

This changes into the xyz coordinate system used by the real-time simulation system from the angle use by (3) and (4). The virtual radius of the celestial sphere can be determined as r, and the z-axis can be set as perpendicular facing north by setting an x-axis, and west into a y-axis, and solar coordinates can be searched for as follows.

$$x = -r * \cos h * \cos A \quad (5)$$

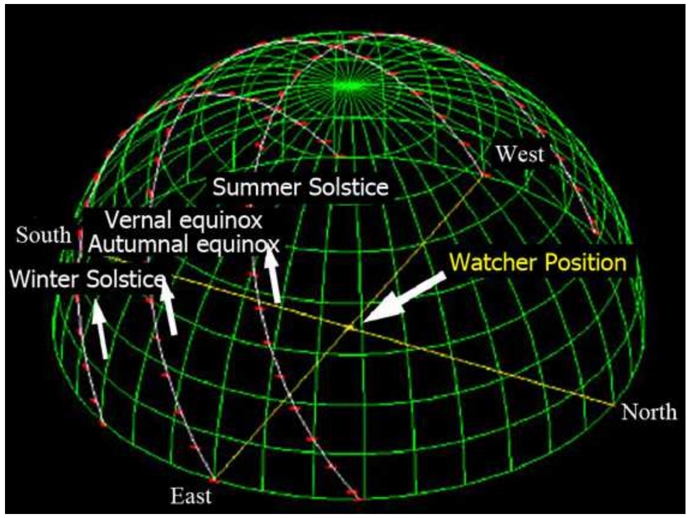

Figure 3: The solar orbit on the celestial sphere (example of Japan)

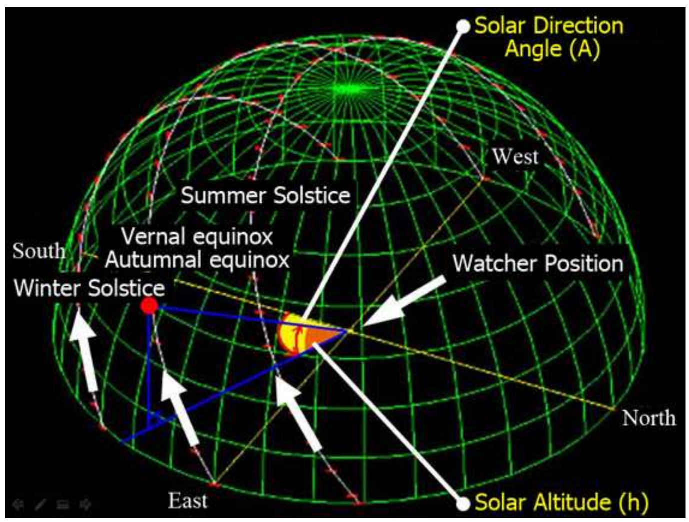

Figure 4: Determination of the solar position on the celestial sphere

$$y = r * \cos h * \sin A \tag{6}$$

$$z = r * \sin h \tag{7}$$

Next, the rendering of the shadow by the projection from light source data is described. There are two methods of carrying out the rendering of shadow in real-time; the *shadow volume method* and the *shadow mapping method*. The generation method of the shadow in the shadow volume method is described. First, the outline of an object is projected from a light source and the shadow domain called the shadow volume, which light does not hit, is created. Next, it is carried out in the process of rendering a shadow on the portion seen from the viewpoint and entering the domain (Fig. 5). On the other hand, by the shadow mapping method, the information on depth distribution of the object seen from the light source is first map-ized per pixel. This map is called a shadow map. Next, when carrying out the rendering of the image from a viewpoint, the pixel which carries out rendering based on the shadow map judges whether it is covered from the light source. A shadow is created when the pixel is covered from the light source (Fig. 6). Since the shadow mapping method performs cover judging for each pixel, it can respond to an object which has been subject to texture mapping with the alpha channel, such as tree data. That is, sunlight shadow expression with high accuracy is possible. However, the shadow mapping method has the fault that calculation speed is slower than the shadow volume method. Therefore, we considered using the shadow volume method for building data, and the shadow mapping method for tree data. The result developed in this section is shown in Fig. 7. The simulation of sunlight shadow is performed in real-time with a simple rectangular parallelepiped object.

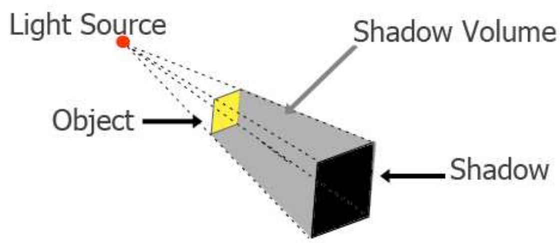

Figure 5: Generation of the shadow by the shadow volume method

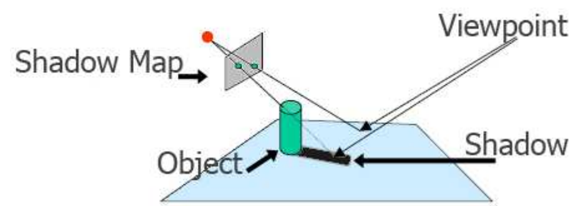

Figure 6: Generation of the shadow by the shadow mapping method

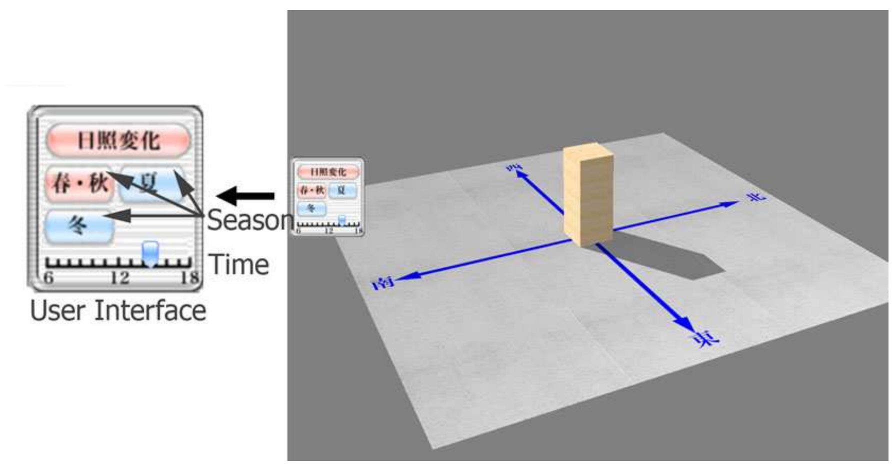

Figure 7: Real-time simulation of sunlight shadow (simple model)

3.2. Grass

The shader programming function of GPU using HLSL (High Level Shader Language) is used, and the expression of grass with textures is developed. The real-time rendering algorithm is the process through which the final surface is output through a vertex input, a vertex shader, a rasterizer, and a pixel shader (Fig. 8). In the *vertex shader stage*, to each vertex which forms a polygon, 2-dimensional coordinate conversion in the output picture and the lighting of each vertex are performed, and the triangular form and triangular vertex color are determined. In the *rasterizer stage*, the pixels inside the triangle formed by the vertex shader are smeared away based on the vertex color. In the *pixel shader stage*, texture mapping and pixel processing of alpha processing etc. are performed on the pixels of the triangle created by the rasterizer, and the rendering of the final surface is carried out. In the shader program used for expression of grass, some special processing routines are added to the real-time

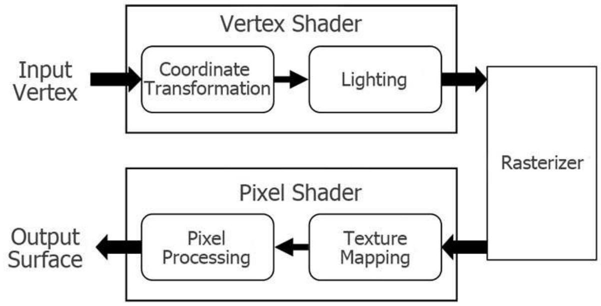

Figure 8: Real-time rendering algorithm

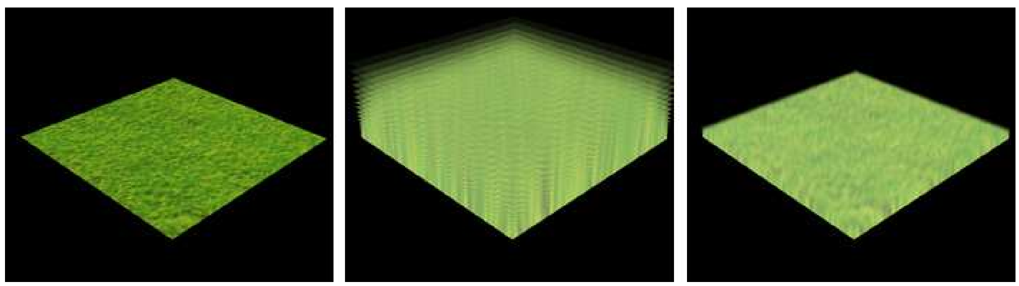

Figure 9: The rendering result by the shader program

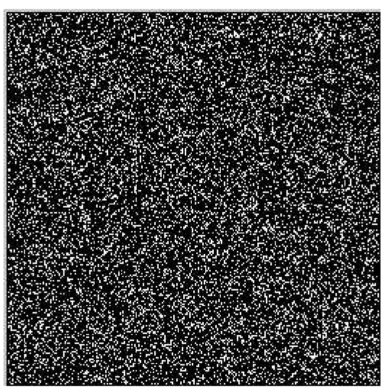

Figure 10: Noise texture

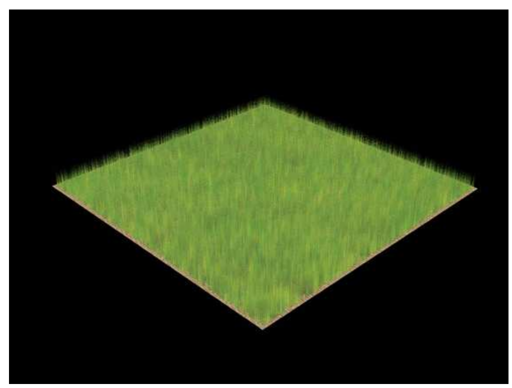

Figure 11: Hair-like rendering

rendering algorithm shown in Fig. 8. The output before adding processing is the expression of a superficial polygon as shown in Fig. 9 on the left hand side. First, in the process of vertex shader coordinate conversion, processing is performed that draws the original polygon up to a screen in piles 20 times in the normal direction of a polygon. Next, in texture mapping of the pixel shader, a texture is mapped to each polygon, raising the alpha value. By this processing, although it was a surface object without the thickness of one sheet in the stage of vertex input, it is thick and the upper part can be displayed as a solid with small density. The interval of the polygons at the time of rendering in piles 20 times is changed, and two kinds of rendering picture of which the solid thickness were changed are shown in the middle and on the right hand side of Fig. 9. Furthermore, in the process of pixel processing, the portion to be rendered can be specified in the shape of a spot by mapping the noise texture shown in Fig. 10. Thereby, an object with the textures of the shape of hair, such as grass, can be drawn (Fig. 11).

3.3. Refraction and reflection on water

It is necessary to consider two elements such as refraction and reflection of light to express a water surface realistically. However, if exact optical calculation of refraction or reflection of

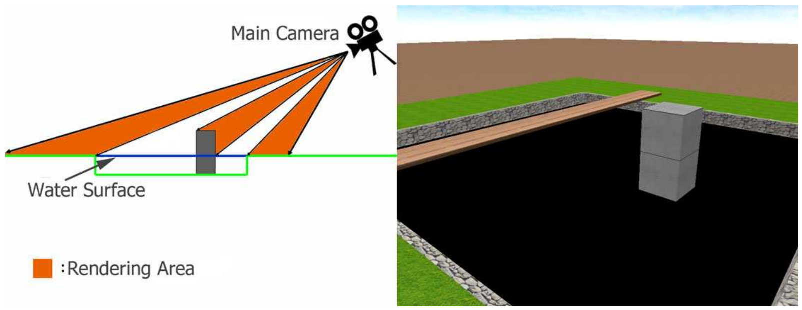

Figure 12: Rendering from output camera

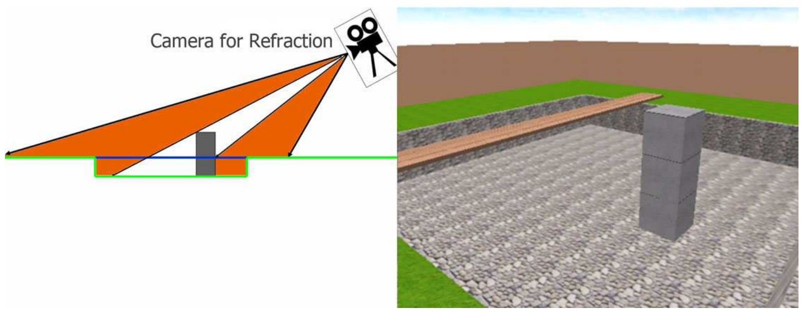

Figure 13: Rendering from refraction camera

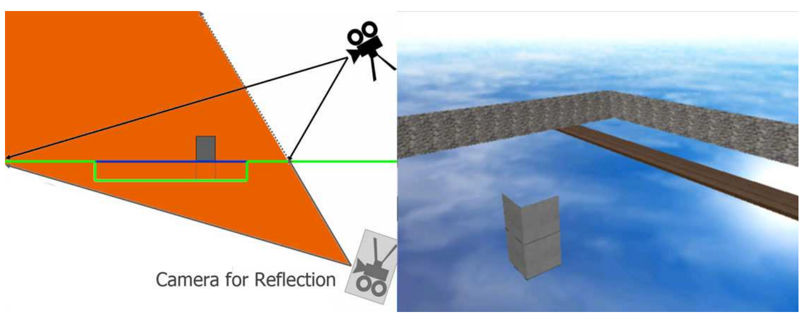

Figure 14: Rendering from reflection camera

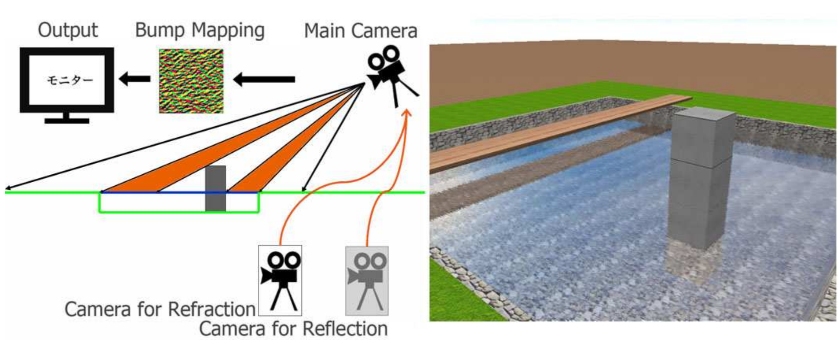

Figure 15: Composition of refraction picture and reflection picture accompanied by bump processing

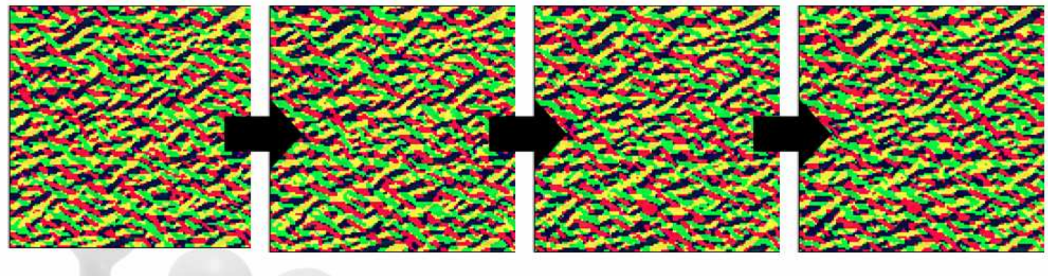

Figure 16: Exchange of bump mapping

light is performed, the rendering of more than 10 fps required for a real-time simulation will become impossible. Therefore, using the GPU rendering method that is currently employed mainly in the field of 3D games, refraction and reflection of light are performed with pseudo-expression and an attempt to produce realistic water surface expression is made. First, from the output camera, the refraction camera, and the reflection camera, the rendering of each image is carried out and saved in a buffer. This is the method of finally piling up and outputting them with bump processing. In overlapping and bump processing of an image, programming processing by the pixel shader described in Section 3.2 is used, and improvement in the speed of rendering processing is attained. The rendering of a scene is first performed from the output camera to the object, except for water surfaces. Thereby, the object picture, except for the water surface, is created (Fig. 12). Next, the rendering of objects in the portion under the water surface is carried out with the refraction camera in the same position as the output camera. This picture is used in the process of bump processing that follows. By this process it is possible to treat a 3D object under the water surface as a 2D picture (Fig. 13). Next, the reflection camera is installed in a reverse position to the output camera at the water surface. The rendering of the object is carried out from this position. Through this process, a mirror-picture for compounding on the water surface with a final output is created. At this time, the object below the water surface is eliminated temporarily and rendering is performed (Fig. 14). The eliminated object is again displayed at the stage at which creation of the mirror-picture ends. Finally, the image of the refraction camera and the image of the reflection camera are laid on top of the water surface portion of the image of the output camera, and bump processing is performed and output. By repeating the above phase, real-time rendering of the water surface including reflection and refraction becomes possible (Fig. 15). In addition, fluctuation of the water surface can be expressed by continuously replacing the bump map used for bump processing (Fig. 16).

3.4. Influence of developed methods on fps

This section evaluates the influence on fps of the expression method developed in Sections 3.1–3.3. Fps is measured when carrying out the rendering of simple contents for a test.

3.4.1. Sunlight shadow of trees

On a ground area of $100\,\mathrm{m} \times 100\,\mathrm{m}$, the contents for the test are placed at 10 meter intervals, and a total of 100 objects with a height of 5 meters are arrange which carried out texture mapping of the trees picture of 256 pixels around. To this object, the source of sunlight is applied from 45°, and the result of having measured the frame rate where a shadow is displayed on the ground (Fig. 17) is shown in Table 2. Since trees need projection in a texture unit, the shadow mapping method was used and the resolution of the shadow map was set as 256 pixels.

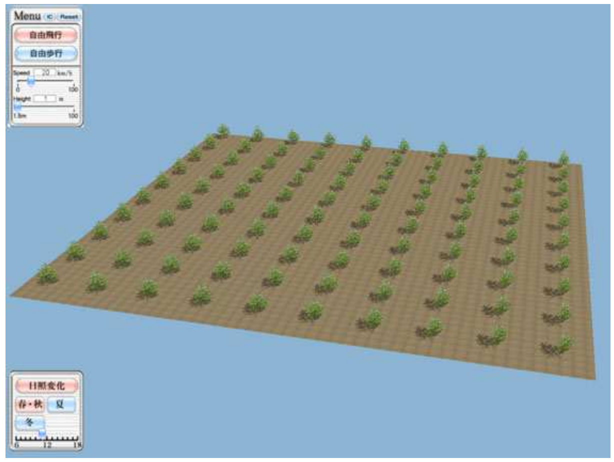

Figure 17: Measurement of fps by shadow display of trees

before shadow display (1)	60.0 fps
after shadow display (2)	16.2 fps
amount of change (2)–(1)	43.8 fps
decreasing rate ((2)–(1))/(1)	73%

Table 2: Change of fps by shadow display

3.4.2. Grass

The contents of the test expressed the shape of hair to the surface of 1 km around which carried out texture mapping of the grass picture of 256 pixels around with the application of the method shown with Section 3.2, and carried out texture mapping of the ground picture of this resolution to it in the surface of the same size. The noise texture used for rendering is shown in Fig. 10, and the resolution is around 256 pixels. The tiling setup of mapping of the ground and grass was set as 2 meters (Fig. 18).

before grass display (1)	60.0 fps
after grass display (2)	48.2 fps
amount of change (2)–(1)	11.8 fps
decreasing rate ((2)–(1))/(1)	19.7%

Table 3: Change of fps by grass display

3.4.3. Water surface

The contents for a test prepare 500 m around and the water surface with a depth of 50 m in the center of the ground of 1 m around. Furthermore, the sky was expressed using a cube

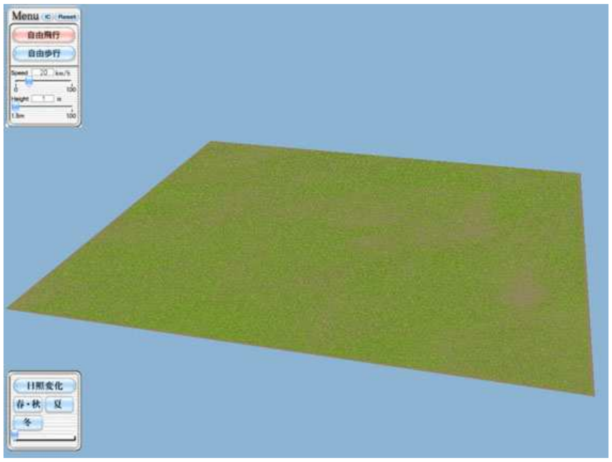

Figure 18: Measurement of fps for grass

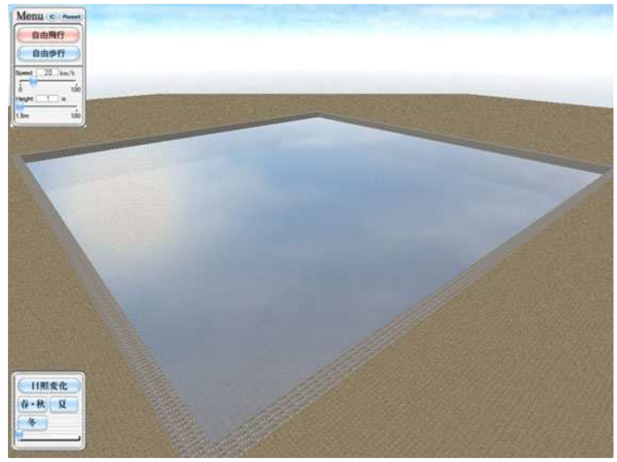

Figure 19: Measurement of fps for water surface

map using a picture of six sheets of around 256 pixels, and it was made to reflect in the water surface, and measured (Fig. 19).

before water surface display (1)	60.0 fps
after water surface display (2)	59.9 fps
amount of change (2)–(1)	0.01 fps
decreasing rate ((2)–(1))/(1)	1.7%

Table 4: Change of fps by water surface

4. Application to environmental design contents, and evaluation

In an environmental design project, the developed expression method needs to evaluate usefulness. Therefore, a park with water surface of about 200 square meters was designed, expression of grass, trees, sunlight shadow, and water surface was performed simultaneously, and change of the frame rate in the rendering from eye level was measured (Table 5). Although a large frame rate comparison change is seen by the display of sunlight shadow, the rate of 10 fps needed for interactive operation also where all expression is applied is securable. This is because the processing in the expression of grass and water surfaces, except for the cover judging of sunlight shadow display, is made by GPU, with hardly any change to the load on the CPU. It became possible to realize natural expression of the same realism as real-time rendering by using the GPU processing function effectively. A rendering picture from each viewpoint of the park contents which applied all the expression methods of grass, water surface, and sunlight shadow of trees is shown in Fig. 20. Moreover, a questionnaire survey was conducted using these contents. This compared the expression method application before and after for a group of 20 persons comprising ten specialists in the environmental design field and ten non-specialists (Fig. 21). Consequently, all 20 persons answered that the direction after application was a realistic expression. Compared with conventional expression, the expression level was better and the opinion that it is easier to understand the completion image of an environmental design was acquired.

Expression situation of each element	*fps*
un-display: grass, water surface, shadow	24.8
display: grass	21.4
display: grass, water surface	21.1
display: grass, water surface, shadow	15.6

Table 5: Change of fps of the park contents by application of the developed expression methods

Figure 20: Real-time rendering pictures of park contents

5. Conclusions and future work

In environmental design, realistic visual expression is needed for all scene composition elements, such as natural elements and artificial elements which surround human beings. Realistic expression can be defined as that which expresses an object on a level corresponding to the accuracy with which a human being perceives a scene. In a close-range view, in order to recognize object details, it is necessary to express the real-time simulation for reviewing an environmental design to a detail. However, real-time simulation of expression which needs rendering of 10 fps was inadequate for close-range view expression of natural elements, which include phenomena accompanied by change while form and color tone are complicated. In this paper, an accurate expression method of sunlight shadow, grass, and refraction and reflection of light on water surface required for a close-range view was developed by effective use of GPU, which specializes in rendering processing. High quality expression that can be

Figure 21: Real-time simulation images for questionnaires. The left picture shows the result before, the right picture after applying the expression method

conventionally used for design review of a close-range view was realized. High quality expression and high speed rendering are realized using HLSL and the shadow volume technique. Each type of software developed in this paper is componentized. Therefore, reuse of other content is possible by applying setting changes. As a future subject, it is necessary to apply the expression method developed in this paper to actual design projects, and to perform more general evaluation. Moreover, it is necessary to develop research into expression methods of natural elements such as rain, snow, and running water and so on which are not treated in this paper.

6. Acknowledgements

This research was done under assistance of the subject number 17686052 of the 2006 Grant-in-Aid for Scientific Research (Japan Society for the Promotion of Science).

References

[1] T. Fukuda, R. Nagahama, A. Kaga, S. Oh, T. Sasada: *Collaboration Support System for City Plans or Community Designs based on VR/CG Technology*. Internat. J. of Architectural Computing bf 1 (4), 461–469 (2003).

[2] W. Yeo, T. Motoi, T. Fukuda, A. Kaga, T. Sasada: *Analysis and Development of Real-time Simulation for Environmental Design*. Proc. 10^{th} Internat. Conf. on Computer Aided Architectural Design Research in Asia **2**, 5–14 (2005).

[3] *Microsoft DirectX9*. http://www.microsoft.com/windows/directx/default.mspx, 2006.

[4] Virtools Company: *Virtools Dev 3.5*. http://www.virtools.com/, 2006.

Received August 7, 2006; final form June 9, 2007