Proceedings of ITC-CSCC 2000, Pusan, Korea

A Simplified Visual Simulation of Urban Space in Consideration with Weather and Sunlight

Rie Kato, Mitsunori Makino
Department of Information and System Engineering, Chuo University
1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan
Phone +81-3-3817-1684, Fax +81-3-3817-1681
{rkatou,makino}@makino.ise.chuo-u.ac.jp

Abstract: In this paper, a simplified visualization method is proposed for an urban space in consideration of weather and sun moving. In the proposed method, buildings and roads with shadows are visualized by the ray tracing algorithm. Also sky, snow, and rain are visualized by textures. Some textures such as snow and rain are generated in advance by the ray casting algorithm. Then we can obtain images with weather condition and shadows of sunlight by buildings along the road in relatively low computational cost.

1. Introduction

Recently, the research on geographic information systems(GIS) [1] has been actively studied and developed. In GIS, visualization tools such as computer graphics(CG) are useful for us to understand several kinds of information related to maps.

The so-called car-navigation system is one of the most popular and useful systems in GIS. The system can show our past and current positions in the map, and also guide a route to our destination. Since such system has to provide necessary and sufficient information whenever we want to know, the visualization in the system must be fast. This is the reason why CG images in the system are not generated usually by many CG models and algorithms which need not so small computational cost. However, present graphics in the navigation system dose not always provide information sufficiently, so that the system sometimes depends on human's understanding. For example, when driver's visibility becomes bad because of shadow and/or bad weather condition, he can not find landmarks in the real world, even if the system shows them in the virtual world. Therefore, it is important to generate CG images of urban space with time-varying information such as weather in low computational cost.

In this paper, a simplified visualization method will be proposed for the urban space in consideration of weather and sun moving. In order to decrease computational cost in visualization, in the proposed method ray tracing algorithm[2] is combined with textures of weather and skylight, which are generated by ray casting algorithm[2][3]. Then we can obtain images with weather condition and shadows of sunlight by buildings along the road in relatively low computational cost.

2. A simplified visual simulation of urban space

In this paper, we assume that the urban data consists of objects such as roads given as a set of meshes and the buildings of which basal plane is a set of meshes. In addition, we assume that the height of each building is invariable(see **Fig.1**) and one of the road is 0. Then we get the data from the map as followings:

- the number of vertexes of each objects
- the coordinate of vertexes of each objects
- the height of each objects
- the color of each objects

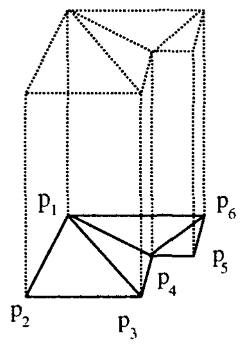

Fig.1. the building data

In this paper, CG images of the urban space are generated by the ray tracing algorithm in consideration of the followings:

- Filtering the urban data
- Shadows by the sun
- Sky and clouds
- Rain and snow

2.1 Filtering the urban data

The computational cost of the ray tracing depends on the number of objects in the space. Usually it is seen that the urban data becomes too much to execute the ray tracing algorithm. In this paper, from the viewpoint and visible area the urban data is filtered, so that the ray tracing can be done with respect to the necessary data(see **Fig.2**). An outline of the filtering is as follows:

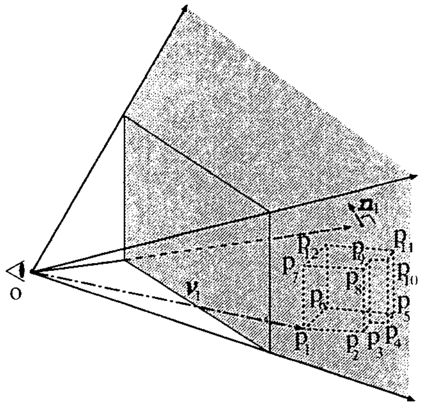

Fig.2. filtering the urban data

Step1: Let n_j, $j = 1, \dots, 4$, be normal vectors of the 4 faces surround the visible area, where direction is inside to the area, and V_i, $i = 1, \dots, N$, be vectors from viewpoint to vertex P_i, $i = 1, \dots, N$, of the building, respectively, where N is the number of vertexes of each object.

Step2: For $i = 1$ to N:

Step2-1: For $j = 1$ to 4:

Step2-1-1: If $V_i \cdot n_j < 0$, go to **Step2** with $i \longleftarrow i + 1$.

Step2-1-2: $j \longleftarrow j + 1$ and go to **Step2-1**.

Step2-2: Since $V_i \cdot n_j \geq 0$ for all j, then it is shown that P_i exists in the visible area. Therefore, the object having P_i is to be visualized, so that it is unnecessary to check other points $P_{i+1}, P_{i+2}, \dots, P_N$.

Step3: Since all P_i, $i = 1, \dots, N$, exist out of the area, the object having P_i is determined as invisible one.

2.2 Shadows by the sun

It is important for high quality car-navigation systems to express shadows by the sun. From time and season, the position and intensity of the sun can be easily obtained (**Fig.3**). Then all shadows of buildings can be generated by solving a set of linear equations derived from the filtered urban data and the position of the sun.

In this paper, we shall divide a year into 4 seasons and decide the position of the sun as follows:

The orbital plane of the sun is vertical as the North Pole in spring or fall. Then the meridian transit vector v_s is given by the outer product between the North Pole vector and the East vector v_E. Because the sun moves by $\frac{1}{12}\pi$rad on it's orbital plane per hour, the position vector v_l of the sun after t hours from sunrise is

$$v_l = v_E + \left|\tan\left(\frac{1}{12}\pi t\right)\right| v_s + v_p. \quad (1)$$

Here, the size of v_s and the radius of the sphere is 1. Also v_p is the vector for parallel translation from the position of the sun in spring or fall to one in summer or winter. Here, the angle of the meridian transit in summer is $\frac{13}{100}\pi$ higher and one in winter is $\frac{13}{100}\pi$ lower than one in spring or fall. From the above, the parallel translation vector v_p can be obtained easily.

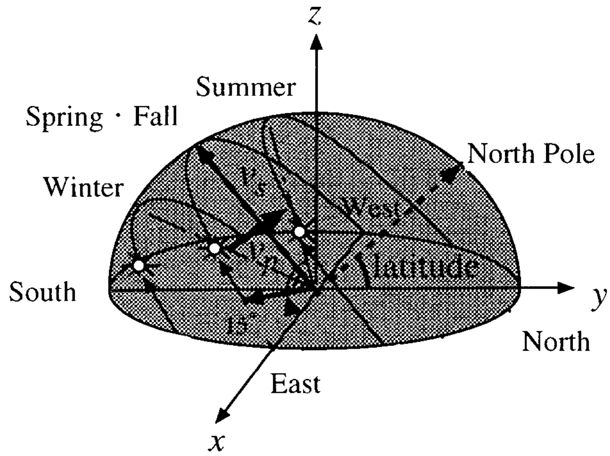

Fig.3. The position of the sun

We regard the sun light as the parallel light. Let L_S be strength of the sun light and θ be the angle of incidence. Then the intensity of the sun light considering with the position of the sun is as followings:

$$L_I = L_S \cos\theta \quad (2)$$

From the above, we can represent the shadows by the ray tracing algorithm, but objects which are far from the viewpoint look dark. Then we shall introduce the depth effect. When we give the distance d which makes the intensity of light reduce by half and define the old intensity of light as L_0, the intensity of light $L(x)$ which is x from the veiwpoint approximate as followings:

$$L(x) = L_0 \exp\left(-\frac{\ln 2}{d} x\right). \quad (3)$$

2.3 Sky and clouds

When a ray intersects no objects, the ray must represent sky. In this paper, the sky and clouds are mapped by a texture, which is selected by the given time, amount of clouds and season. Here, the intensity of the skylight L_I is calculated from the amount of clouds:

$$L_I = (1 - 0.1 A_C)\, L_S. \quad (4)$$

Here, L_S is the intensity of the fine sunlight. A_C denotes the amount of clouds in the sky, which varies from 0 to 10 according to the weather[4]. The textures of the clouds are real pictures.

2.4 Rain and snow

Rain and snow should be considered because they influence our visibility very much. In this paper, textures of rain or snow is located in front of all objects in the urban space. The texture is selected by the given weather condition. We use some textures of rain or snow so that we can express the difference of the amount of rain or snow. In this paper, textures of rain are generated by some drawing tools and the ones of snow are generated by the ray casting algorithm to give a lot of small balls in the voxel space. Also we represent mist under rain by using texture, which is generated by the ray casting algorithm. In addition, we can consider the road and top of the building which is wet and covered with snow by changing the attribution of the road.

2.5 Outline of the proposed algorithm

From the above, we propose the following algorithm:

Step1: Calculate the intersection point by the ray tracing algorithm.

Step2: If the intersection point of **Step2** is on a road or building, calculate the intensity of the sunlight L_I from the position of the sun and the amount of the clouds. Set the reflectivity from the normal vector and weather. Then calculate the intensity from the intensity of the sunlight and reflectivity. If necessary, trace again to the direction of reflection.

Step3: If the intersection point of **Step2** is on the sky, map the texture of the sky.

Step4: If the weather is rain or snow, map the texture of it.

From the above, CG images of the urban space can be generated by only linear computation with weather condition as well as time and season. Since the data needed complex calculation is a priori obtained as a set of textures, the main process of rendering images is simplified.

3. Simulation

For example, geographic information near Korakuen campus, Chuo University, is used for all simulation, which has 24 objects with 214 meshes. In all simulation, images are generated from a viewpoint to the east of the campus. **Table 1** ~ **Table 3** show computational time for each simulation under the 440MHz Ultra SPARC IIi system. The size of all images is 400 × 300 pixels.

Table 1. fine day in spring

Figure	season	weather	time	computing time[s]
Fig.4	spring	fine	7:00	131.47
Fig.5	spring	fine	12:00	129.88
Fig.6	spring	fine	17:00	113.12
Fig.7	spring	fine	20:00	100.45

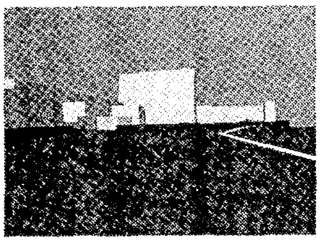

Fig.4. at 7:00

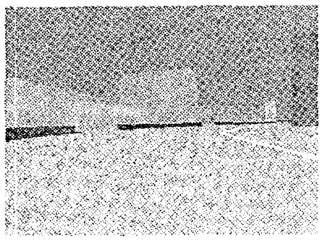

Fig.5. at 12:00

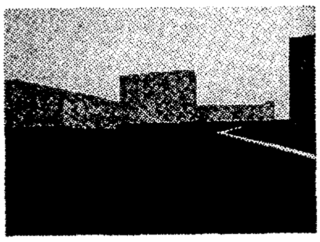

Fig.6. at 17:00

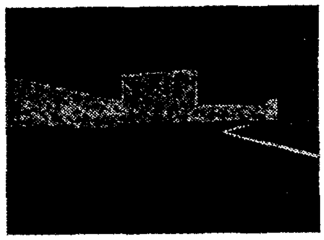

Fig.7. at 20:00

Fig.4~ **13** show 10 examples of the proposed method There exist 21 objects and 184 meshes in the visible area, and we caluclate only these objects for rendering. **Fig.4**~ **7** are the simulations in the same day at the different time. In **Fig.4**, shadows on buildings by the sun is observed and they correspond to the real world. **Fig.5** shows that the ground is bright, because the sun is high, and shadows by the sun is moving from the position of shadows in **Fig.4**. **Fig.6** shows the sky dyed red by sunset and the light is beginning to fail. Because the position of the sun is low, we change the texture of the blue sky into one of the red sky. When time past from **Fig.5** to **Fig.7**, the intensity of the light becomes lower.

Table 2. Difference in amount of clouds

Figure	season	amount of clouds	time	computing time[s]
Fig.8	fall	2	10:00	129.98
Fig.9	fall	4	10:00	133.64
Fig.10	fall	6	10:00	134.95
Fig.11	fall	8	10:00	99.76

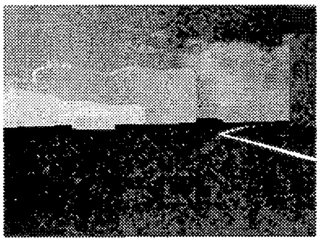

Fig.8. Amount of clouds is 2

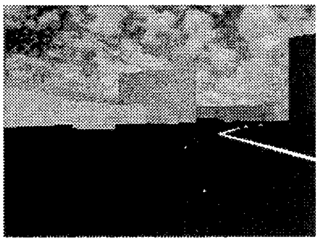

Fig.9. Amount of clouds is 4

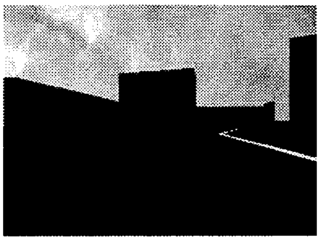

Fig.10. Amount of clouds is 8

Fig.11. Amount of clouds is 10

Table 3. Rainy day and snow day

Figure	season	weather	time	computing time[s]
Fig.12	summer	rainy	10:00	102.57
Fig.13	fall	snow	10:00	99.81

Fig.8 ~ **Fig.11** are cases in consideration of the amount of clouds. When the amount of clouds increases, we see the intensity of light becomes lower.

Fig.12 is the case of rainy day and **Fig.13** is the one of snow day. They show their visibility becomes worse than fine day at the same time (**Fig.8**).

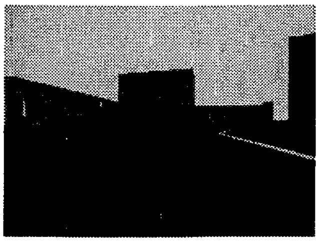

Fig.12. Rainy day at 10:00 in Summer

Fig.13. Snow day at 10:00 in Winter

4. Conclusion

In this paper, we have proposed a simplified visual simulation for the urban space in consideration with weather condition and sun moving. The proposed method consists of 2 parts; pre-process for generating textures and main process for rendering. Consequently, it is seen that the computational cost for the main part becomes relatively low in spite of representing shadows and weather. While we can't consider shadows by the removed objects. We shall consider shadows by the removed objects so that the generated images are realer.

In a future paper, we shall consider other effects such as moonlight. In addition we shall use the textures of the building's design so that CG images of the urban space are realer.

Acknowledgement

This research is partially supported by the Research Project on "Integrated Geographic Information Systems", and on "High Quality Solid Modeling and Visualization in Computer Graphics", at the Institute of Science and Engineering, Chuo University.

References

[1] Gorge B.Korte, "THE GIS BOOK; Understanding the value and Implementation of Geographic Information Systems, 4th ed.", OnWord Press, 1997.

[2] T.Whitted, "An improved illumination model for shaded display ", Communications of the ACM, vol.23, no.6, pp.343-349, 1980.

[3] A.Kaufman, "Volume Visualization", IEEE Computer Press, 1993.

[4] S.Miyazawa, "Weather Map and Weather Book ", Kokusai-Tigaku-Kyoukai, 1993.