A Modeling and Rendering Method for Snow by Using Metaballs

Tomoyuki Nishita, Hiroshi Iwasaki

Fukuyama University
Sanzo, Higashimura-cho, Fukuyama, 729-02 Japan

Yoshinori Dobashi

Hiroshima University
1-4-1, Kagamiyama, Higashi-hiroshima, 739 Japan

Eihachiro Nakamae

Hiroshima Institute of Technology
2-1-1 Miyake Saeki-Ku, Hiroshima, 731-51 Japan

Abstract

The display of natural scenes such as mountains, trees, the earth as viewed from space, the sea, and waves have been attempted. Here a method to realistically display snow is proposed.

In order to achieve this, two important elements have to be considered, namely the shape and shading model of snow, based on the physical phenomenon. In this paper, a method for displaying snow fallen onto objects, including curved surfaces and snow scattered by objects, such as skis, is proposed. Snow should be treated as particles with a density distribution since it consists of water particles, ice particles, and air molecules. In order to express the material property of snow, the phase functions of the particles must be taken into account, and it is well-known that the color of snow is white because of the multiple scattering of light.

This paper describes a calculation method for light scattering due to snow particles taking into account both multiple scattering and sky light, and the modeling of snow.

Note: *snow, multiple scattering, Mie scattering, metaball, volume rendering*

1. Introduction

Snow plays an important role in the displaying of natural scenes in winter, ski simulators, computer games, and commercial films. Commercial software able to display snowfall has been developed. In this paper, we attempt to display scenes containing snow fallen onto objects and scattering snow. Generally, conventional methods have not been able to generate realistic images.

Though snow is displayed in the same way as ordinary objects are in previous work, for making more realistic images we should consider the scattering effect due to particles in the snow. That is, this paper discusses not only a local illumination model, but also a global illumination model taking into account both the color variation of incident light as it passes through the atmosphere and sky light. In order to display realistic snow, computational model has to closely follow the physical phenomenon. Volumetric representation is suitable, since snow consists of water particles, ice particles, and air molecules. This paper proposes a volumetric modeling method for the shape of the snow and a shading model for the material property of snow.

As for modeling of snow, the following properties are taken into account.

1) Snow falls onto objects and lies along the surfaces. In particular, snow on curved surfaces is taken into account. 2) Density distribution of snow varies depending on its weight (density is higher in deeper part of the snow). 3) Scattered snow falls according to the Newton's law.

The shape and density distribution of snow is modeled using metaballs. Uneven snow surfaces are generated by distributing metaballs along curved surfaces of the objects.

As for the shading, the following points are considered. Snow consists of small crystals of ice, and the color of snow is determined by scattered light due to particles within it.

To display realistic images, a precise shading model is required: two components should be considered. One is multiple scattering due to particles in the snow: the albedo of snow is very high: It is well known that for objects with such a high albedo, multiple scattering can not be ignored [4]. The other factor to be considered is sky light. That is, snow is illuminated by both direct sunlight and sky light affected by atmospheric scattering. For the former, the calculation of snow intensities has been assumed to be complex due to strong forward scattering. However, this paper proposes an efficient calculation method using these scattering characteristics in a positive way.

The authors have developed a method for displaying clouds taking into account multiple anisotropic scattering [23] and the proposed method is an extension of the method to the displaying of snow.

Section 2 contains the basic idea of the proposed method, section 3, the modeling method, section 4, the shading model, and in section 5, several examples are demonstrated in order to show the effectiveness of the method proposed here. Finally, in section 6, conclusions are presented.

2. Basic Idea and Previous Work

A brief description of our proposed method is as follows. The proposed model can treat three different types of objects, polygons, Bezier surfaces (i.e., parametric surfaces), and metaballs (i.e., implicit surfaces) which are often used for representing animals, clouds, and so on. The proposed method can display snow on objects'surfaces that are expressed by any of these.

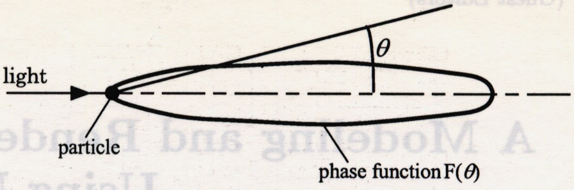


Figure 1: *Phase function for Mie scattering.*

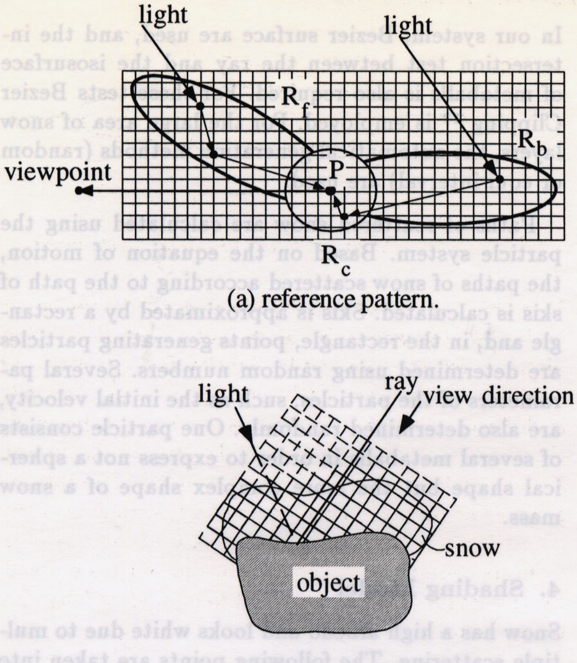
In this paper, the density distribution of the snow is defined by using the meta-ball technique [21] (or blobs [2]). Snow is defined by density fields, which are modeled by the metaball technique. Each metaball is defined by its center, radius, and the density at the center of the ball. The field value at any point is defined by distances from the specified points in space. The density distribution of a metaball is given by a polynomial function in degree 6 of the distance from its center [21]. The surface of the snow is defined by the isosurfaces of potential fields defined by the set of metaballs.

In the rendering process, the intersection of the isosurface (i.e., the snow surface) with the viewing ray is calculated by ray tracing, which effectively expresses the density distribution on a ray by using Bezier function of degree 6 (see [21]). Bezier Clipping [19] is employed for the calculation of intersections.

The intensity from snow reaching the viewpoint is determined by light scattered and absorption due to particles in the snow. In order to render the particles in snow, the following elements should be taken into account: (i) Phase functions should be taken into account; scattering by small particles such as air molecules is called Rayleigh scattering, and scattering by particles is called Mie scattering. The sizes of particles in snow are relatively large (i.e., 2-40 mm), so they have strong forward scattering (see Fig.1). (ii) The multiple scattering of light among particles in snow can not be neglected because their albedos are very high [4] [26]: 0.85 for new snow. (iii) The snow is illuminated by both direct sunlight and sky light. Sunlight is absorbed when light passes through the atmosphere. (iv) The density distributions of snow are not uniform.

The display of snow is a kind of volume rendering. We have to take into account scattering due to particles. There are various scattering effects in natural phenomena such as fog, clouds, fire, dusty air, sky color, water color, shafts of light in atmosphere/water, smoke, and skin color. There are a lot of previous work considering scattering effects.

Such previous work on single scattering is as follows: a) light scattering from particles in the air [18] [8] [16], b) the sky color due to atmospheric scattering



(a) reference pattern.

(b) voxel including snow.

Figure 2: *Shading model for multiple scattering.*

[14 11 20], c) clouds and smoke [18 25], d) the color of water such as ponds [12 20] and optical effects such as shafts of light within water [22], and e) Saturn's rings (reflective ice particles) [2].

The previous work taking into account multiple scattering is as follows: a) the radiosity of a participating medium [24], b) multiple anisotropic volume scattering [9 16 1], c) subsurface scattering such as skin [7], and d) gaseous phenomena such as fire [27].

In this paper we focus our discussion on multiple scattering. We have proposed a display method of clouds which takes into account multiple scattering, and sky light effects [23]. The proposed method is extended version of it.

The particles have strong forward scattering characteristics. This has been considered as a problem in the previous work due to intensity calculation. We approach this in a positive way. That is, the space to be calculated is restricted because the scattering direction is very narrow (see Fig.2(a)). For the calculation of multiple scattering, the space containing the snow is subdivided into a number of volume elements (voxels; see Fig.2(b)). As a preprocess, a sample space is prepared, which is defined as a parallelepiped consisting of a set of voxels with the average density of the snow. Then the high order of scattering at a specified voxel from the other voxels in the space is calculated and stored, before the calculation of scattering due to every voxel in the total space. By using this pattern, which is the contribution ratio at each voxel in the sample space to the specified voxel, the calculation cost for the total space can be reduced. At least the 3rd order of scattering is calculated in our paper.

3. Modeling of Snow

Snow consists of snow flakes(crystals of ice), air molecules, and water particles. It is represented not as a surface but as a volume with density distribution. Not only snow on the ground, the shape of which is virtually flat, but also snows on objects expressed by curved surfaces are taken into account. Snow scattered by, for example, skis is also taken into account.

In our previous paper, the complicated cloud surfaces are generated by a fractal technique applying to metaballs [23]. Three characteristics of the geometric model of snow are different from that of clouds: i) the surface of snow is relatively smooth in most cases, ii) the outline is determined by the shape of objects covered in snow: snow always covers the tops of objects. iii) the density for snow depends on its height from the object surface. The density at the bottom is high because of the weight of the snow particles, and the rate of air gaps is higher near the surface. Also the shape of the fallen snow flakes near the surface is not distorted.

We propose a modeling method expressed by metaballs; the snow surface is defined by the isosurface of the field function, and density distribution of particles in snow is also defined by the field density due to metaballs.

We should create the snow surface along the object surface; in some cases the object surface is curved. In our system, the curved surfaces are defined by Bezier patches. The center of the metaball is set at the surface of objects; through this the density at the bottom of the snow becomes high. Along the surfaces many balls are set and compose a layer, and if necessary a second layer is generated by covering some of the metaballs on the isosurface of the metaballs by using the polygolization technique [13]. By repeating this process we can get any depth of snow and can control the smoothness of the snow surfaces: If we create metaballs with a narrow interval, we can get a smooth surface because an isosurface of close metaballs is relatively flat. Conversely, if we create metaballs with a large interval or random intervals, we can get uneven surfaces.

For rendering snow grains, small primitives are used. Ice grains in snow are not necessarily spherical, in most case the shape is a six-sided prism when

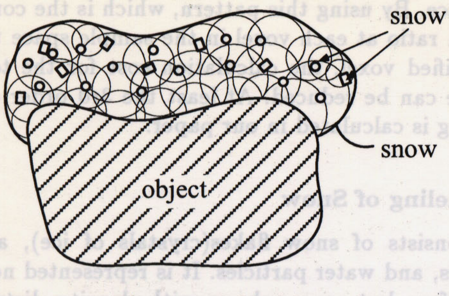


Figure 3: *Modeling of snow.*

just having fallen onto an object. But as snowflakes change with time after they settle, to be assumed this shape as spherical: we call this sphere a sub-ball because the shape of the snow is modeled using metaballs. And the density distribution in the sub-ball is assumed to be the same as that of the metaball. Relatively new snow grains are assumed prisms (approximated by 4-sided), That is, in this model, very small primitives (i.e., sub-balls/prisms) exist within the metaballs which define the snow surface (see Fig. 3). The number of these sub-balls is very large, so the sub-balls are defined by solid texture. This is similar to Kajiya's texell mapping [10]. The sub-balls are used in a shallow region. As shown in Fig. 3, snow is modeled as a multiple structure, that is, very small metaballs, representing particles, are included in metaballs that define the density distribution. The distribution of small primitives are calculated as follows. Sub-balls (prisms) are randomly generated in the sub-space and their positions and sizes are stored as look-up tables. It is assumed that this distribution pattern is repeated to fill the whole space of snow.

Snow falls from the sky and comes to lie on objects (free-fall) or on the side of objects (due to wind). In this paper, snow on curved surfaces is generated interactively instead of the physical simulation. Therefore, metaballs are placed on the surfaces at first and new balls are added if necessary. The new balls are generated on an isosurface. Surfaces on which snow lies can viewed from the direction from which the snow falls. In most cases, snow lies on the tops of objects, even though it attaches to the sides of the objects in some cases because of drifting. We developed an interactive system for creating snow models. After displaying the top view of the scene with hidden surface removal, we can easily specify the positions of metaballs by clicking arbitrarily points on the visible surfaces. By using an inverse mapping technique, transformation from the screen coordinates to the world coordinates , we can get the 3-D position on the surface. This inverse mapping can be solved by intersection tests between the ray passing through the clicked pixel and the surfaces. In our system, Bezier surface are used, and the intersection test between the ray and the isosurface of metaballs is also required. For these tests Bezier Clipping [19] is employed. For the large area of snow layers, the automatical generation methods (random or equi-interval) are used.

Paths of scattered snow are calculated using the particle system. Based on the equation of motion, the paths of snow scattered according to the path of skis is calculated. Skis is approximated by a rectangle and, in the rectangle, points generating particles are determined using random numbers. Several parameters of the particles, such as the initial velocity, are also determined randomly. One particle consists of several metaballs in order to express not a spherical shape but the more complex shape of a snow mass.

4. Shading Model

Snow has a high albedo and looks white due to multiple scattering. The following points are taken into account. 1) The phase function is taken into account (since the scattering due to snow particles obeys the Mie scattering theory, the particle has strong forward scattering as shown in Fig. 1). 2) Multiple scattering due to both direct sun light and sky light is taken into account. Multiple scattering is calculated up to the third order.

Snow on objects is one common example of a multiple scattering medium. The calculation method is similar to that for clouds [23], that is, snow is also illuminated by direct sunlight and sky light. The differences are as follows. Snow consists of ice grains, air, and water droplets. The representative value of ice grains in snow is 1mm; they are much larger than cloud droplets. That is, the phase function is more narrow (asymmetry factor g is 0.93 [4]; see reference [23] for g). Because deep snow is optically thicker than clouds, snow is brighter.

By applying a method described in [23], multiple scattering is calculated as follows. Let's denote the intensity at point P in direction ω as $I(P,\omega)$, the extinction coefficient per unit length as ρ, the length of snow in viewing ray S, the path length from P as s ($s = 0$ at P, P_s;S from P). Then $I(P,\omega)$ is expressed by

$$I(P,\omega) = I(P_s,\omega)\exp(-\tau(PP_s)) + \int_{s=0}^{S} [$$

$$\beta\rho(s)\exp(-\tau(P(s))P)\frac{1}{4\pi}\int_{4\pi} F(\theta)I(P,\omega^{'})d\omega^{'}]ds$$

where θ is phase angle between ω and ω'. F is phase function (see Fig. 1). The problem is that I exists in both sides of the equation. To solve this, this space is subdivided into a number of volume elements. If we denote the number of voxels as N, and the number of the discrete directions as M, then MN matrix equations should be solved.

The space containing snow is divided into voxels and shooting/receiving energy among them is calculated. In the computation, we make use of the fact that voxels with a high contribution to a certain voxel can be limited to voxeles within a certain area by taking into account the property of the phase function and form factors between voxels. That is, a sample space is prepared, that tells us which voxels contribute to the voxel. Fig. 2(a) shows distribution of voxels which have high contribution to the viewing direction. Three sub-spaces R_f, R_b, and R_c, contribute to the light scattered at voxel P in the viewing direction. The 1st order of scattering due to particles in R_f is strong because of the small phase angle, even though the 2nd order of scattering at P is weak because of the large phase angle. Even though the 1st order scattering in R_b is weak because of a large phase angle, the 2nd order of scattering at P is strong because of the small phase angle. In R_c, distance of voxels from P are very close, so the form factors are large even though the phase functions are small. We call the sample space a reference pattern. Using the reference pattern, the scattered light reaching the viewpoint from all voxels can be calculated efficiently.

The main difference between snow and clouds is the size of the ice particles, as mentioned above. Although small metaballs are taken into account where the density is low, they are neglected where the density exceeds a certain threshold since their influence is very small. Two types of small primitives exist in the proposed method: sub-balls represented by metaballs and prisms for the calculation of specular reflection. The flat surface of a prism often reflects sunlight like a mirror, causing specular glints in snow, which we can observe. The presence of snow grains should be considered and grains near the surface can not be ignored. The normal vectors of prism surfaces are defined randomly.

Fig.4 shows the optical paths for the shading of snow. The intensity of each pixel can be determined by integrating the scattered light along the viewing ray through the pixel. The intensity reached at the viewpoint is determined by the single scattering at particle P due to sunlight and sky light, multiple scattering (e.g., path $P_1 P_2 P$), and reflected light from object surface P_a. If the ray hits any sub-balls, their densities are added: the intensity of scattered light depends on the snow density. Otherwise, if the ray hits the prisms, the component of specular reflection is calculated and the ray tracing process is terminated.

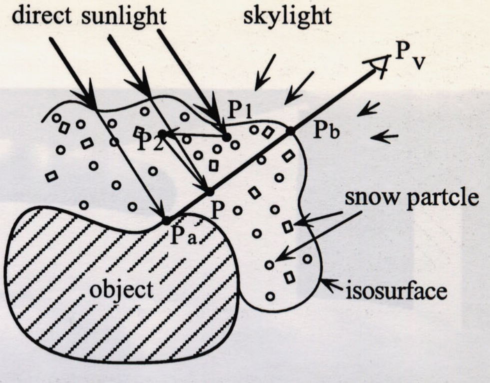


Figure 4: *Optical paths for shading of snow.*

Even though the intensity at the viewpoint can be obtained by the integration of scattered light on the viewing ray ($P_b P_a$ in Fig. 4), the integration is processed from P_b to P_a. As optical depth of snow gets thicker, the bottom is often not visible.

In most cases, integration can be stopped before reaching the bottom when the optical depth from the snow surface exceeds the given value and sub-balls (to represent snow grains) in the deeper region can be neglected. Thus in the integration process, the distribution of snow grains is taken into account up to the specified optical length.

For sky light, the spectrum of the sky light is calculated by taking into account scattering/absorption due to particles in the atmosphere, and the intensity of light scattered at one voxel illuminated by each sky element is stored. That is, the spectrum and spatial distribution of sky light are precalculated by taking into account both Rayleigh scattering and Mie scattering by assuming negligible attenuation due to cloud particles. The intensity of the first order scattering at each voxel due to sky light can be easily calculated by using the optical depths from the snow surface stored in a look-up table.

5. Examples

Fig. 5* shows a car covered in snow. The car and the trees are modeled by Bezier patches and the respective number of metaballs are 1088 for snow and 782 for clouds. To display clouds multiple scattering is also taken into account by the method [23]. Figs6. (a), (b), (c), and (d) are close ups of Fig. 5. These are a comparison of rendering methods; (a) snow surfaces are displayed as opaque surfaces (iso-density surfaces defined by meta-balls), (b) single scattering

* See pages C-397 and C-398 for Figures 5 and 6.

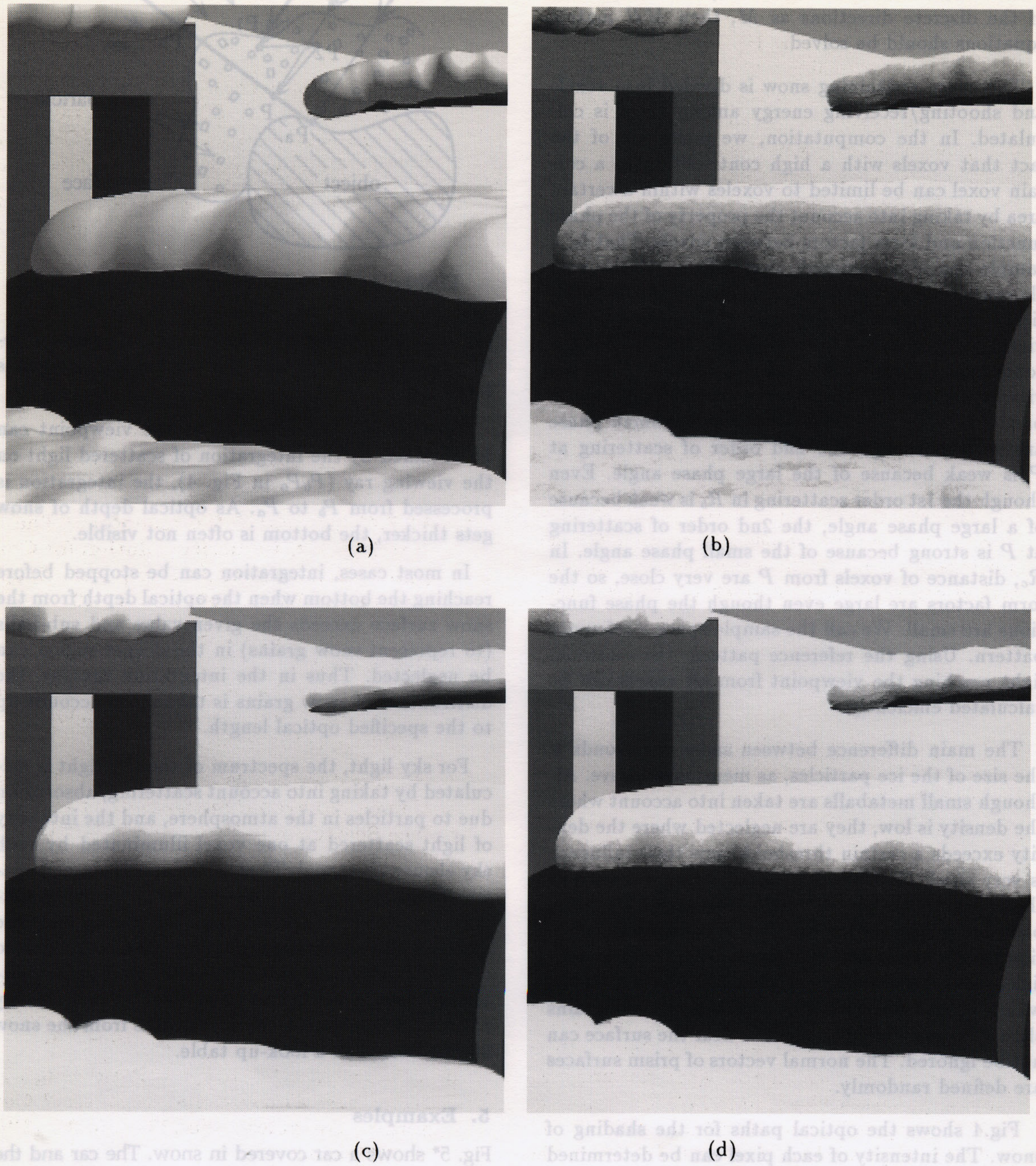

(a)
(b)
(c)
(d)

Figure 6: Comparison of rendering methods.

with sub-balls, (c) multiple scattering without sub-balls, (d) multiple scattering with sub-balls. These figures show the effect of multiple scattering (comparing with Figs. (a) and (d)) and effects of sub-balls (comparing with Figs. (c) and (d)). For rendering this image, a modified raytracing method [21] is used, in which both parametric and implicit surfaces (i.e. metaballs) can be rendered without polygonization. Shadows cast by snow are calculated (see the bottom right in Fig. 5), even though shadows due to parametric surfaces are ignored. This calculation was done on an IRIS Indigo2. The computation time for Fig. 5 is 28 minutes (image width = 500).

Figs. 7(a) (b)* show scenes including a mountain and a skier. The close mountain and far mountains are represented by Bezier surfaces (40 patches) and polygons (77,939 polygons created by Fractal) respectively. The skier consists of 16,739 polygons. For snow 7,563 metaballs are used. In this example, snow scattered by skier's skis is also displayed. The computation time for Fig. 7(a) is 90 minutes.

In the examples, the size of voxels is 110^3, as in Fig.5, so we can save the memory by using the list-structure. As shown in these examples, the proposed method can generate photo-realistic images including snows.

6. Conclusion

We have proposed an algorithm for a physical based image synthesis of clouds and snow. As shown in the examples, the proposed method gives us photo-realistic images taking into account anisotropic multiple scattering and sky light. The advantages of the proposed method are as follows:

1. The snow can be modeled by using metaballs. Smooth surfaces of snow covering objects can be generated by setting metaballs along the top surfaces of the objects.
2. For anisotropic multiple scattering, the optical paths of the light scattered in the viewing direction are limited because of strong forward scattering (a narrow phase function). Employing the pattern expressing the contribution ratio at each voxel to a specified voxel in the sample space, the calculation cost for the total space can be reduced.
3. For rendering the effect of snow particles, structured metaballs are proposed, with small metaballs called sub-balls and small prisms distributed within the larger metaballs. The sub-balls define the variation of density due to snow grains, and the prisms can be used for specular reflection due to particles in freshly fallen snow.

Acknoledgment

The authors wish to thank Mr. S. Yamaa for his help in producing the models.

References

1. **P. Blasi, B.L. Saec, C.Schlics, "A Rendering Algorithm for Discrete Volume Density Objects", Proc. of EUROGRAPHICS'93, Vol.12, No.3 (1993) pp.201-210.**
2. **J.F.Blinn, "Light Reflection Functions for Simulation of Clouds and Dusty Surfaces," Computer Graphics, Vol.16, No.3 (1982), pp.21-29.**
3. **J.F.Blinn, "A Generalization of Algebraic Surface Drawing," ACM Transaction on Graphics, Vol.2 (1980) 235**
4. **C.F. Bohren, "Multiple Scattering of Light and Some of its Observable Consequences, " Am. J. Phys., Vol.55, No.6 (1987) pp.524-533.**
5. **M.F.Cohen and D.P.Greenberg, "The Hemicube: A Radiosity Solution for Computer Environment," Computer Graphics, Vol.19, No.3 (1985) pp.31-40.**
6. **W.M.Cornette, J.G.Shanks, "Physical Reasonable Analytic Expression for the Single- Scattering Phase Function," Applied Optics, Vol.31, No.16 (1992) pp.3152-3160.**
7. **P.Hanrahan, W.Krueger, "Reflection from Layered Surfaces due to Subsurface Scattering," Proc. of SIGGRAPH'93 (1994) pp.165-174.**
8. **M.Inakage, "Volume Tracing of Atmospheric Environments," The Visual Computer, (1991) pp.104-113.**
9. **J.T.Kajiya, B.V. Herzen, "Ray tracing Volume Densities," Computer Graphics,Vol.18, No.3 (1984) pp.165-174.**
10. **J.T.Kajiya, Y.L.Kay, "Rendering Fur with three Dimentional Textures," Computer Graphics, Vol.23, No.3 (1989) pp.271-280.**
11. **K.Kaneda, G.Yuan, E.Nakamae, T.Nishita, "Photorealistic Visual Simulation of Water Surfaces Taking into Account Radiative Transfer," Proc. of CG & CAD'91 (China) (1991) pp.25-30.**
12. **K.Kaneda, T.Okamoto, E.Nakamae, T.Nishita,"Photorealistic Image Synsesis**

* See page C-399 for Figure 7.

for Outdoor Scenery Under Various Atmospheric Conditions," The Visual Computer, Vol.7 (1991) pp.247-258.

13. W.E.Lorensen, H.E.Cline, "Marching Cubes: a High Resolution 3D Surface Construction Algorithm," Computer Graphics, Vol.21, No.4 (1987) pp.163-169.

14. R.V.Klassen, "Modeling the Effect of the Atmosphere on Light," ACM Transaction on Graphics, Vol.6, No.3 (1987) pp.215-237.

15. N.Max, "Light Diffusion through Clouds and Haze," Graphics and Image Processing, Vol.33, No.3 (1986) pp.280-292.

16. N.Max, "Efficient Light Propagation for Multiple Anisotropic Volume Scattering," Proc. of the Fifth Eurographics Workshop on Rendering (1994) pp.87-104.

17. T.Nishita, and E. Nakamae, "Continuous Tone Representation of Three-Dimensional Objects Illuminated by Sky Light," Computer Graphics, Vol.20, No.4 (1986) pp.125-132.

18. T.Nishita, Y.Miyawaki, E.Nakamae, "A Shading Model for Atmospheric Scattering Considering Distribution of Light Sources," Computer Graphics, Vol.21, No.4 (1987) pp.303-310.

19. T.Nishita, T.W.Sederberg, M.Kakimoto "Ray Tracing Rational Trimmed Surface Patches," Computer Graphics, Vol.24, No.4 (1990) pp.337-345.

20. T.Nishita, T. Shirai, K.Tadamura, E.Nakamae, "Display of The Earth Taking into Account Atmospheric Scattering, " Proc. of SIGGRAPH'93 (1993) pp.175-182.

21. T.Nishita and E.Nakamae, "A Method for Displaying Metaballs by Using Bezier Clipping," Proc. of EUROGRAPHICS'94, Vol.13, No.3 (1994) c271-280.

22. T.Nishita, E.Nakamae, "Method of Displaying Optical Effects within Water Using Accumulation Buffer," Proc. of SIGGRAPH'94 (1994) pp.373-379.

23. T. Nishita, Y. Dobashi, E. Nakamae,"Display Method of Cloud Taking into Account Multiple Anisotropic Scattering and Sky Light", Proc. of SIGGRAPH'96 (1996) pp.379-386.

24. H.E. Rushmeier, K.E. Torrance, "The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium," Computer Graphics, Vol.21, No.4 (1987) pp.293-302.

25. G.Sakas, M.Gerth, "Sampling and Anti-Aliasing of Discrete 3-D Volume Density Textures," Proc. of Eurographics'91 (1991) pp.87-102.

26. S.Sekine,"Corrected Color Temperature of Daylight(2) : Characteristics on Clear Sky and Overcast Sky," J. Illumination Engineering Inst. Japan, Vol.79, No.11 (1995) pp.621-627.

27. J.Stam, E.Fiume, "Depicting Fire and Other Gaseous Phenomena Using Diffusion Processes," Proc. of SGGGRAPH'95 (1995) pp.129-136.