Snow Accumulation and Generation

Yuanhui Chen

Abstract

This project is to create the snow accumulation and falling effects. The method used for snow accumulation is very like the shadow mapping. The snow falling effect is created by a simple particle system.

Keywords: OpenGL, Snow Accumulation

1 Previous work

Before this method, there are many ways to create beautiful snow scenes. But most algorithms cannot generate snows in real time. Paul Fearing [2000] presented "Computer Modelling of Fallen Snow". It worked by first tracing snow paths from the ground upwards towards the sky and then accumulate the snow. In Hsu and Wong' paper "Visual Simulation of Dust Accumulation", they use an exposure function to tell wheter the surface is in dust or not. The exposure function is calculated by sampling around area with rays.

2 Overview

This project is based on technique report from NVIDIA. There are basically 4 paths. Frist path is to render a depth map into a texture in framebuffer. Second path is to render the scene with depth map. Third path is for snowflakes rendering. The final path is to add fog effect.

3 Snow Accumulation

The process is just like the shadow mapping.

Step1. Create a depth map from the view of sky. This view is an orthographic projection. From this path, the depth buffer is extracted and saved in this frame. This map will be saved as textures in graphics memory. This map stores the closest points to the sky in the scene. As long as the scene doesn't change, we do not need to recreate the depth map.

Step2. Render scene from the camera view with the depth map. First, find the coordinates of the vertices in the sky view coordinates system. Secondly, compare the depth of the vertices with stored depth. If less than stored vertices, then points can be occupied by the snow, else the points are occluded. Use the following matrices to for orthographic view projection:

VL = Ms * Mp * Mv * Mm * V

VL projected vertex to light source projection

Figure 1: *Depth map.*

Ms scale matrix (-1, 1) to (0, 1)

Mp light source projection

Mv light source view from world

Mm model to world

V vertices

Step3. Incline Coefficient

Slopes will accumulate less snow than flat ground, so this is a need to add an incline coefficient to represent this. To accomplish this, simply use the dot product of surface normal with Y axis. Use the incline coefficient to blend the object color with snow color. From the following picture, we could see that the top of the sphere accumulate more snow than lower part.

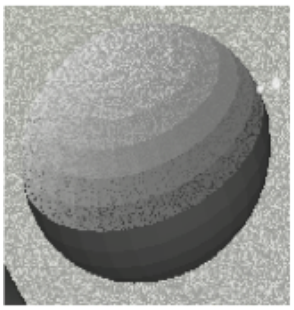

Figure 2: *Without Coefficient*

4 Snow Generation

Use particles to represent snowflakes. Randomly generate speed, size, and positions for each particle. Then animate particles with vertex shader. For the realistic effects, I also calculate the alpha channel that the further points are more translucent. Instead of drawing points several times, I use draw points and point sprites features in OpenGL. In such way, all particles could be drawn at one time.

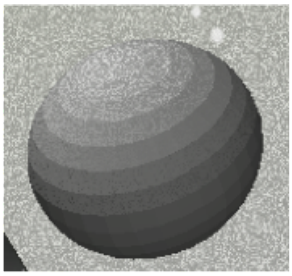

Figure 3: *With Coefficient*

5 Fog Effect

The fog effect is added in the fragment shader. First, for each pixel, calculate it's distance from the camera, and then use an exponential equation to get the fog factor. Finally, use this fog factor to blend real colors with fog colors.

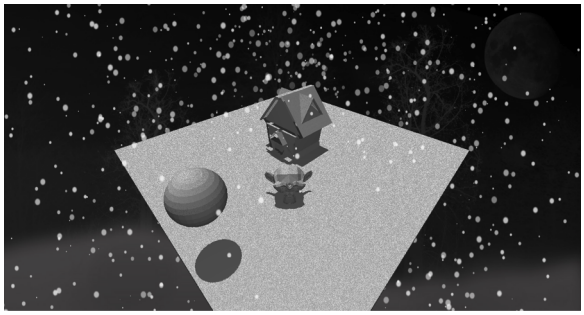

Figure 4: *Without Fog*

6 Result

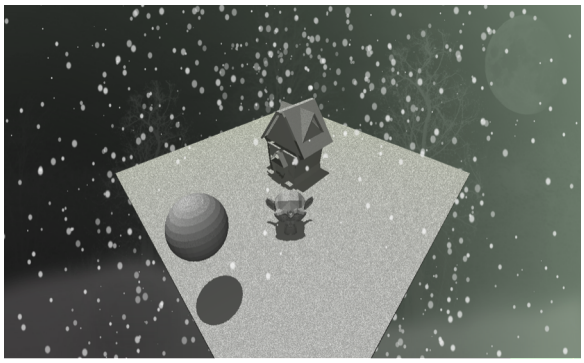

Figure 5: *Final Result*

From this picture, we could see that some areas where are under the shadow of other objects are not occupied by snows. There are also fogs in the pace. Snowflakes far away from us are more transparent compared with near ones.

7 Future Work

1. Cannot create the height of snow accumulation. This method just uses texture for snow effects. And whether points are occluded are only determined by the depth value from orthographic view. 2. This method fails in situations where there are winds. In such situations, points may be not exposed to the sky, but they may also be occupied by snow. 3. Snow only moves in the vertical action. It will be better if I could add winds. 4. Export this to WebGL.

Reference

Nvidia Technical Report Snow Accumulation

Rendering Particles

Tutorials: Fog in GLSL

Tutorials: Shadow mapping