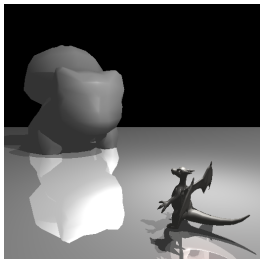
Final Project

Weiqiang Liu Hanyu Xue

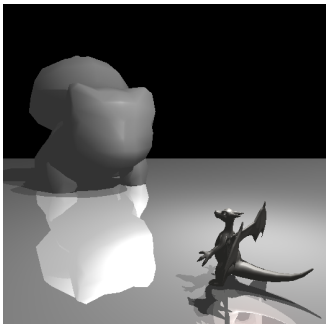
Mesh Intersection



Model is composed of thousands of triangles. Mesh intersection actually is to combine triangle intersection and z-buffer. The way to implement triangle intersection is to calculate the intersection point of the ray and the plane the triangle in. Then judge whether the point is in the triangle.

Texture Mapping

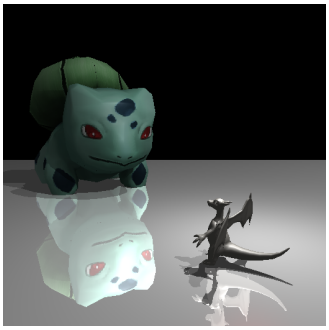
To add texture to an model, we provide an png image as texture and an mesh model. Each triangle of the mesh have an (u,v) pair, which responds to the coordination in the texture image. The algorithm calculates (u,v) for each pixel using



interpolation.

Soft Shadow

Consider point light as area light. Using an random cell to change point light to other position, calculate the hitting state, average the color value.



Without Soft Shadow

With Soft Shadow (49)

With Soft Shadow(100)

Texture Tiling

We use the texcoord of the intersection as (u, v) using the interpolation. When (u, v) is inside the range[0, 1], we can just look up the value in the texture.

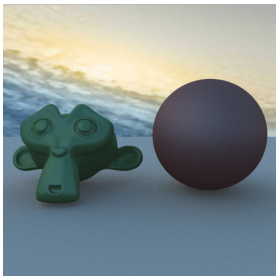
When (u, v) is outside the range [0, 1], they can be calculated by the following equation.

$$u = \begin{cases} u - (int)u & if\ u > 0 \\ u - (int)u + 1 & if\ u < 0 \end{cases} \qquad v = \begin{cases} v - (int)v & if\ v > 0 \\ v - (int)v + 1 & if\ v < 0 \end{cases}$$

Environment Illumination

Environment illumination is how the surface reflects the environment light. It can be computed by the following two formulas.

$$C_l = L\ *background\ /\ pdf$$



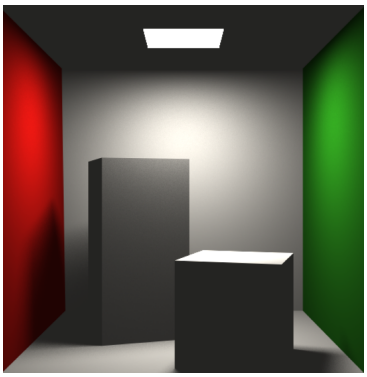
- pdf is the second value returned by the built-in function sample_brdf
- background is the attribute value looked up in the background image.

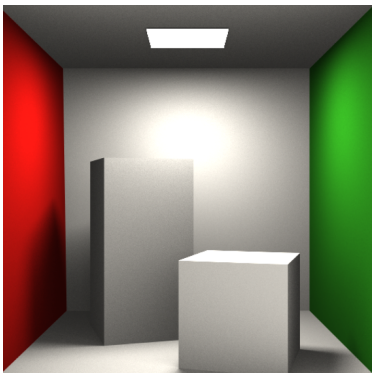
$$L = brdf\ *cos$$

- brdf is bidirectional reflectance distribution function used to characterize the directional properties of how a surface reflects light
- cos can be computed by the dot value of direction and normal.

Indirect Illumination

We will sample the brdf for indirect illumination and pick direction and pdf, compute the material response (brdf*cos) and accumulate recursively do the path tracing.

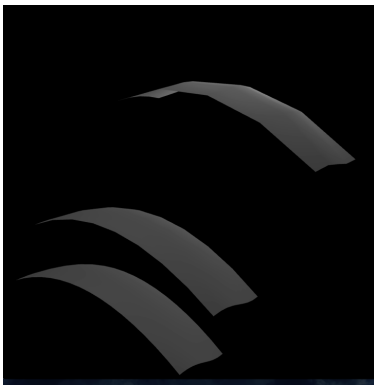




Bezier Surface Patches

Each row of four control points can be considered as a 2D Bezier curve. We will do uniform sampling to get the new control points for each row and use these control points as 2D Bezier curve to do again for each column. Thus, a point on a 3D Bezier surface can be defined as the weighted sum of all of the control points in the patch.

$$Q(u, v) = \sum_{i=0}^{3} \sum_{j=0}^{3} P_{ij} B_i(u) B_j(v)$$



Demo Scene

We create a scene of pokemon, which is a manga of Japan. Two characters model is put in the scene. They are fighting are the ground which has high reflection value. The owner of one of them is at background to witness the battle. We also add environment illumination into the scene.

Brief-assessment

We think that the most difficult and rewarding part is mesh intersection and bezier surface patches. We used the model from the Blend website and exported it to json file. We applied our rendering part to the model and got the demo scene, which is extremely interesting.

Reference

1. Model fron Blend Swap Website: http://www.blendswap.com/blends/view/6545
2. Model from Blend Swap Website: http://www.blendswap.com/blends/view/79263