Weizheng Liang
COMP3271
Prof. Taku Komura
11/24/2022

Written Assignment 2

(Note: all figures are from lecture note of this course)

1. (20 marks)
(a) Show the difference between perspective projection and orthogonal projection methods by discussing their transformation matrices and rendering results.

Ans:

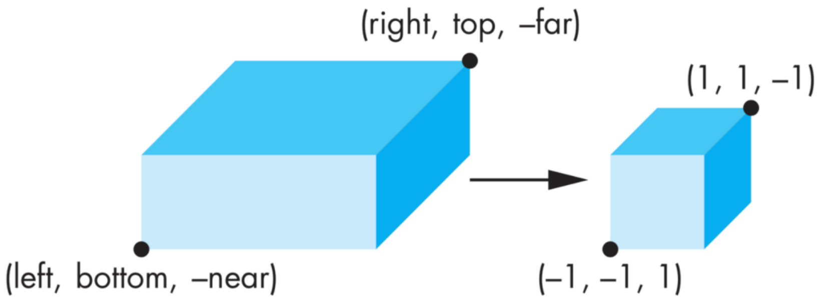
Projection is to put a 3D object onto a 2D plane. As we can imagine that we draw straight lines(projectors) from every point on the object to a plane.

Orthogonal projection is a method that each line that is originally parallel will be parallel after the projection, while the center of projection is replaced by a direction of projection. We will see the direction of projection is the same as the center of projection at infinity.

Perspective projection allows all projectors meet at the center of projection (toward the camera), and the lines that are originally parallel are no longer parallel. Instead, the parallel lines converge to a point where the camera at. More like a real-world view in this case.

Regarding the transformation matrix of these two projection methods, since most graphics systems use view normalization, all (most) views are converted to the canonical view by transformations that determine the projection matrix, including the two projection methods we are talking about here. So, we shall know the difference in their transformation matrix by talk about how a specific clipping volume (we can think of it as a volume formed by the object plane, projection plane and projectors) being normalized and transformed into a canonical volume in these two methods.

Orthogonal projection: This one is not difficult. Since all projectors are parallel to each other, we can imagine the shape of the clipping volume is some cuboid.



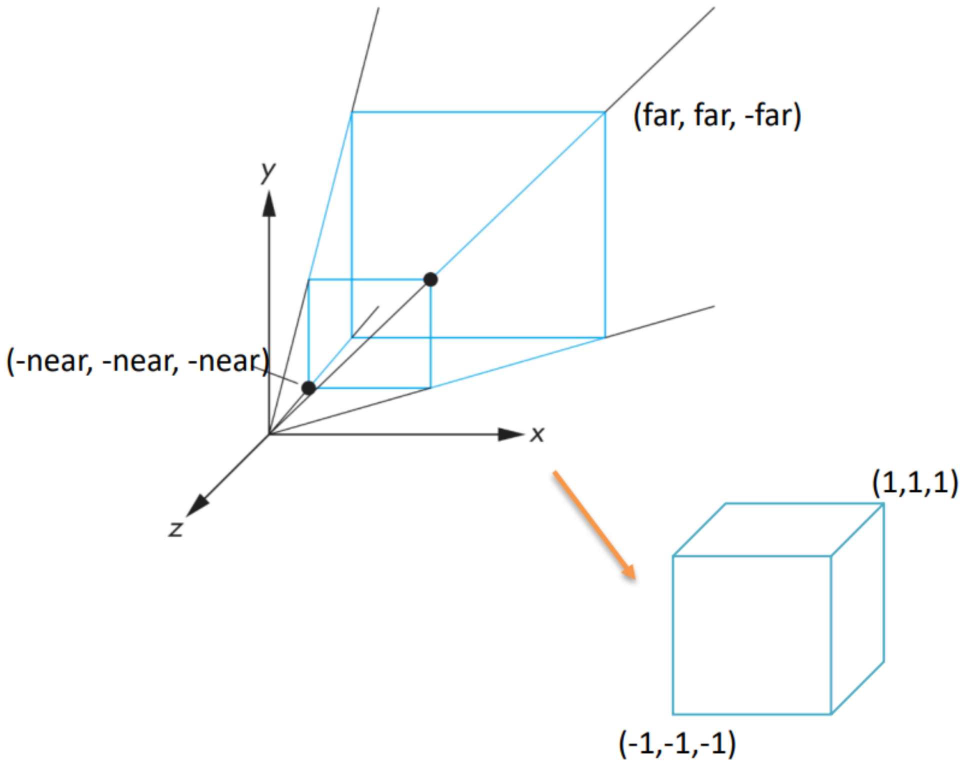
As shown in above figure, we shall do two steps to do the transformation.

1. Move center to origin
2. Scale to have sides of length 1-(-1) = 2

Hence, we shall have the transformation matrix as: **translation** + **scale**

$$\mathbf{P} = \mathbf{ST} = \begin{bmatrix} \frac{2}{right-left} & 0 & 0 & -\frac{right+left}{right-left} \\ 0 & \frac{2}{top-bottom} & 0 & -\frac{top+bottom}{top-bottom} \\ 0 & 0 & \frac{2}{near-far} & \frac{far+near}{far-near} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Perspective projection: Similarly, in this method, we also transform the clipping volume to canonical view volume.



We can see the clipping volume looks like a frustum this time. It not easy to do the transformation this time. One approach here is to first transform the frustum to a cuboid that looks like the orthogonal projection clipping volume. Then transform it to canonical view volume. We already knew the latter step which exactly the same as the orthogonal one.

Therefore, we have M(persp) = M(ortho)M(persp->ortho)

(b) Describe the Z-buffer algorithm briefly, and explain how it is used for hidden surface removal.

Ans:

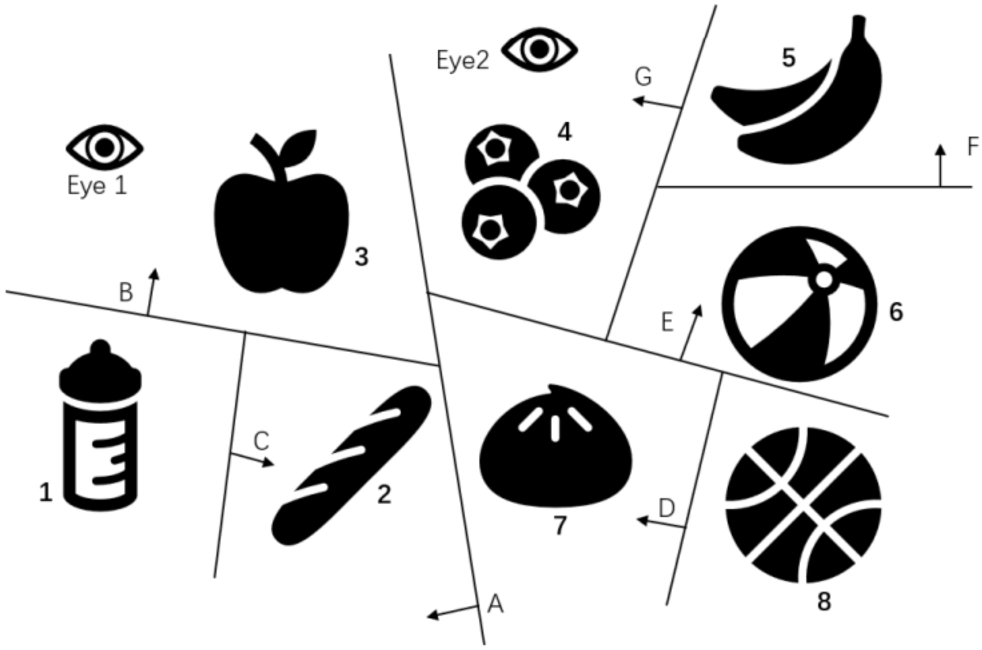
Z-buffer algorithm is used to store floating-point depth information for each pixel in the frame buffer.

Before a fragment's color is written into the frame buffer, we shall

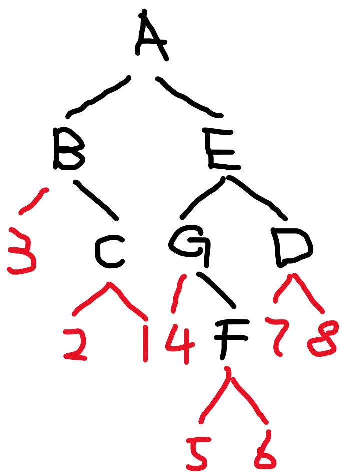
1. Compares its depth with the value stored at the corresponding pixel of the depth buffer.
2. If the depth is smaller than the existing value, overwrites the pixel in the frame buffer and update the depth buffer with new depth value
3. Otherwise, discard the fragment.

It solves the problem where painter algorithm has. Order now does not matter, but the depth. This algorithm only draws what's on the surface, which automatically discards or removes the hidden surface.

2. (20 marks) (a) Consider a scene with 8 objects (numbered 1 to 8) as shown in the figure below. Construct the corresponding BSP tree. (b) Based on the BSP tree, give the back-to-front rendering sequence of the objects from the viewpoints Eye1 and Eye2, respectively.



Ans: (a)



(b) Eye1: {8,7,6,5,4,2,1,3}; Eye2: {1,2,3,8,7,6,5,4}

3. (20 marks) (a) Describe the three basic components of Phong illumination equation; (b) Describe how the Flat shading method and Gouraud shading method work; (c) Describe the pros and cons of Phong shading method.

Ans:

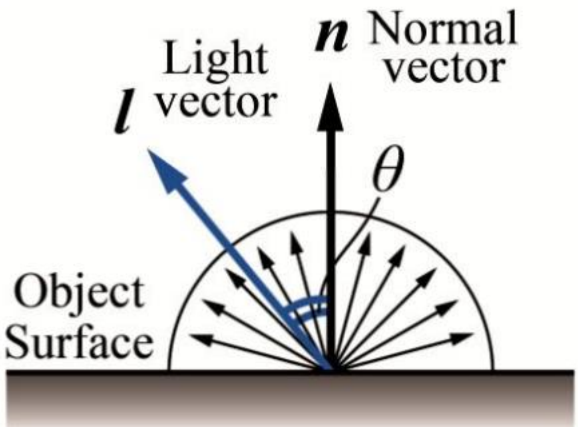
(a) The three basic components of Phong illumination equation are:

i. Diffuse - $I_L k_d (n \cdot l)$

ii. Specular - $I_L k_s (r \cdot v)^n$

iii. Ambient - $I_a k_a$

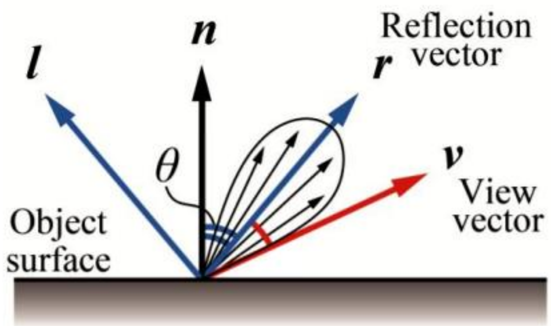
Diffuse: Diffuse Reflection is the simplest kind of reflection. It models dull, matte surfaces, some materials like chalk. Ideally, it scatters incoming light equally in all directions. The appearance of this reflection is identical from all viewing directions.



Equation: $I_L k_d (n \cdot l)$

I_L is the light source intensity
k_d is the diffuse surface reflectance coefficient, $k_d \in [0,1]$
θ is the angle between normal & light direction

Specular: Specular Reflection will form a shiny look, or glossy specular highlights, because they reflect light coherently in a preferred direction. For the same reason, their appearance changes as the viewpoint moves.



We can see that the light reflected about normal direction. And we also see the reflected rays pointing directions to form a cone area about the reflection vector. This is because most surfaces are imperfect specular reflectors. One exception is mirror, which is a perfect specular reflector, it's reflected ray pointing only in the direction of reflection vector but nothing in any other direction.

Equation: $I_L k_s (r \cdot v)^n$

I_L is the light source intensity
k_s is the specular surface reflectance coefficient, $k_s \in [0,1]$
ϕ is the angle between viewing and reflection direction
α is the shininess factor (approaching to infinity corresponds to perfect mirror, 5 to 10 is something look like plastic)

Ambient: Ambient Light is to work on the areas that are not directly illuminated by any light, which will appear black. This ambient light will make our material look more natural. We shall just make up a constant ambient glow, which is purely fictious, just to make it natural.

Equation: $I_a k_a$

I_a is the ambient light intensity
k_a is the ambient surface reflectance coefficient

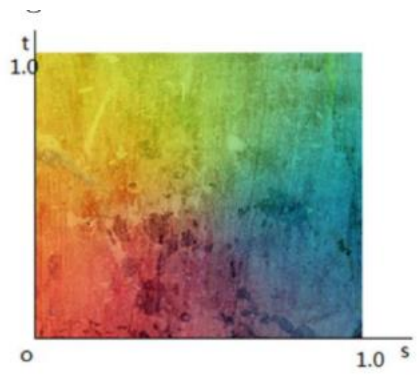
(b) Flat Shading and Gouraud Shading

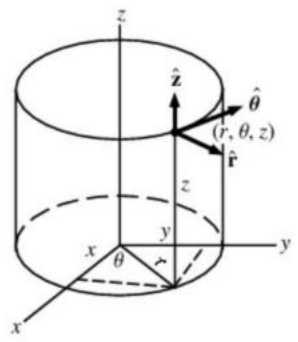
a) **Flat Shading**: constant color
It is basically a shading model that compute the normal vector of each face of the polygon (do lighting calculation for every polygon), then to compute the color per polygon. At the end, to fill every pixel covered by polygon with the resulting color.
However, this shading model suffers from "Mach band effect". It a type of artefact makes humans sensor sudden change of brightness. This artefact remains even though the polygon number is increased.

b) **Gouraud Shading**: interpolate vertex colors
Instead of calculating lighting and color for every polygon, in this shading model, we do them once for each vertex. We compute color for each covered pixel and linearly interpolate colors over polygon. We can also use barycentric coordinates to compute interpolate the color for every pixel center.
However, Gouraud shading is not good when the polygon count is low. In this case, misses details that don't fall on vertex. May lead to artefact.

(c) **Phong shading**: interpolate vertex normals
This time we go even further. In this method, we want to do lighting calculation for every pixel covered by a triangle. This method has the best shading effect but computationally intensive.

4. (10 marks) Consider a texture pattern defined in a unit square [0, 1]2 and a cylindrical surface S, as depicted in the figure below.





The cylindrical coordinates of a point on S can be represented as (r cos ✓, r sin ✓, z), where r is the radius of the cylindrical surface and 0 ✓ 2↑, 0 z 1. Suppose we want to wrap the texture on S. Derive a mapping which assigns a texture coorinates in [0, 1]2 to a point on S.

Ans:

First, need to map the rectangular bitmap to this cylinder. Any point on cylinder can be computed by:

1. $(r\cos\theta, r\sin\theta, z)$
2. $0 < \theta < 2\pi$
3. $0 < z < 1$

then texture map coordinates mapped to cylinder via:

1. $(u, v) = \left(\frac{\theta}{2\pi}, z\right)$
2. $0 \leq u, v \leq 1$

5. (10 marks) Describe the pros and cons of screen-door transparency compared with alpha-blending.

Ans:

Alpha-blending is good but expensive as requires sorting.

Screen door transparency, compared to alpha-blending, does not required sorting. However, there are issues when overlapping objects with the same alpha. Must do this in the screen space – not in the object space.

6. (20 marks) (a) Explain the problem produced by using projective shadows on planes, and show how stencil buffer can help solve the problem; (b) Show the failure case of shadow volume method, and explain why it fails in this case; (c) Describe the cons of shadow map method compared with shadow volume and explain the reason.

Ans:

(a) Problem:

Since alpha blending is implemented to make the shadow semi-transparent, shadows sometimes overlapping with each other which creates darker or light areas. Also, the shadows are at the same level of the floor. They usually get into z-fighting.

Solution by Stencil Buffer:

For the overlapping issue, Stencil buffer can let some fragments through but not others which can prevent the overlapping part of the shadow from projected multiple times. To do this:

1. Set the stencil on for the pixels of the floor
2. Only render the shadows on pixels that the stencil is on
3. Once a shadow is rendered, the stencil value is set to zero
4. No need to lift the shadows above the floor
5. Shadows not rendered outside the floor

(b)

(c) The precision is worse than the shadow volume. The shadow volume can calculate the intersection of shadows and objects. And problem also occurred if the resolution of the shadow map is too low. Aliasing artefact.