Assignment 3: Ray Tracing Basics

NAME: BINGNAN LI
STUDENT NUMBER: 2020533092
EMAIL: LIBN@SHANGHAITECH.EDU.CN

1 INTRODUCTION

(1) Generate ray from camera
(2) Ray geometry intersection
(3) Phong lighting at intersection points
(4) Soft shadow
(5) Anti-aliasing via super-resolution
(6) Texture mapping
(7) Normal and displacement texture
(8) Advanced anti-aliasing via rotation

2 IMPLEMENTATION DETAILS

(1) Generate ray from camera

Once we get coordinate of pixel $(dx,\ dy)$, we can calculate the corresponding coordinate in world space by the following procedure:

$$2 \times \frac{dx}{x_{resolution}} - 1 = \frac{x_{coef}}{AspectRatio \times focal\_length \times \tan(fov/2)}$$
$$2 \times \frac{dy}{y_{resolution}} - 1 = \frac{y_{coef}}{focal\_length \times \tan(fov/2)}$$

Then the world coordinate is

$$P = \mathbf{forward} + x_c oef \times \mathbf{right} + y_c oef \times \mathbf{up}$$

And the direction of generated ray is

$$\mathbf{ray\_dir} = normalize(P - P_{camera})$$

Thus, the expression of ray is

$$r(t) = P_{camera} + t \times \mathbf{ray\_dir}$$

(2) Ray geometry intersection

(a) Triangle intersection:

Assume that ray is $r(t) = o + t\mathbf{d}$ and barycentric coordinate of intersection point is $(b_1, b_2, 1 - b_1 - b_2)$, then we have

$$o + t\mathbf{d} = (1 - b_1 - b_2)p_0 + b_1 p_1 + b_2 p_2$$
$$o + t\mathbf{d} = p_0 + (p_1 - p_0)b_1 + (p_2 - p_0)b_2$$

Let $s = o - p_0,\ e_1 = p_1 - p_0,\ e_2 = p_2 - p_0$, we can rewrite the equation as following form:

$$[-d, e_1, e_2]\begin{bmatrix} t \\ b_1 \\ b_2 \end{bmatrix} = s$$

Then, by Cramer's rule, we have

$$\begin{bmatrix} t \\ b_1 \\ b_2 \end{bmatrix} = \frac{1}{|-d\ e_1\ e_2|}\begin{bmatrix} | & s & e_1 & e_2 & | \\ | & -d & s & e_2 & | \\ | & -d & e_1 & s & | \end{bmatrix}$$

Then let $s_1 = d \times e_2,\ s_2 = s \times e_1$, we have

$$\begin{bmatrix} t \\ b_1 \\ b_2 \end{bmatrix} = \frac{1}{s_1 \cdot e_1}\begin{bmatrix} s_2 \cdot e_2 \\ s_1 \cdot s \\ s_2 \cdot d \end{bmatrix}$$

Once we get b_1 and b_2, we can check whether $b_1 \geq 0, b_2 \geq 0$ and $b_1 + b_2 \leq 1$, if they do, then ray intersects with triangle.

(b) Rectangle intersection:

The known knowledge of rectangle is its geometrical center p_0, length x, width y, normal vector $\mathbf{n}$ and tangent vector $\mathbf{t}$.

Based on that, we can firstly calculate cotangent vector $\mathbf{c}$ by cross product:

$$\mathbf{c} = normalize(\mathbf{n} \times \mathbf{t})$$

Then we calculate the intersection point p

$$n(o + t\mathbf{d} - p_0) = 0$$
$$t = \frac{n \cdot p_0 - n \cdot o}{n \cdot \mathbf{d}}$$
$$p = o + t\mathbf{d}$$

Then we check whether

$$\mathbf{pp_0} \cdot \mathbf{t} \leq \frac{x}{2}$$
$$\mathbf{pp_0} \cdot \mathbf{c} \leq \frac{y}{2}$$

if they do, then ray intersects with rectangle;

(c) Ellipsoid intersection:

The know knowledge of ellipsoid is

- C = Center of the ellipsoid in world space
- $\mathbf{a}$ = First ellipsoid axis vector (along local x-axis)
- $\mathbf{b}$ = Second ellipsoid axis vector (along local y-axis)
- $\mathbf{c}$ = Third ellipsoid axis vector (along local z-axis)
- L_0 = Point on the line in world space
- $\mathbf{v}$ = Vector that defines the line direction in world space
- P = Point on the surface of a unit radius sphere centered in the origin

Computer Graphics I Course Assignment

First, we need to transfer ellipsoid into a standard unit sphere. In order to do that, we need to construct the following three matrices:

$$T = \begin{bmatrix} 1 & 0 & 0 & C_x \\ 0 & 1 & 0 & C_y \\ 0 & 0 & 1 & C_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} \hat{a}_x & \hat{b}_x & \hat{c}_x & 0 \\ \hat{a}_y & \hat{b}_y & \hat{c}_y & 0 \\ \hat{a}_z & \hat{b}_z & \hat{c}_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$S = \begin{bmatrix} ||a|| & 0 & 0 & 0 \\ 0 & ||b|| & 0 & 0 \\ 0 & 0 & ||c|| & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M = TRS$$

Then we use M^{-1} to transform ray

$$r'(t) = M^{-1}o + tM^{-1}\mathbf{d} = o' + t\mathbf{d}'$$

Then check if $r'(t)$ intersects with sphere, we can determine whether $r(t)$ intersects with ellipsoid.

(3) Phong lighting at intersection points and soft shadow

For each intersection point, we construct shadow ray to perform visibility test. The construction process is like: for each light sample, generate new ray from intersection point to light sample. After that, we need to check whether this shadow ray intersects with other geometries between intersection point and light sample. If visibility succeed, then we apply phong lighting model into that point. Once we traverse all sample light, we divided diffuse and specular light result by sample size to get correct value.

(4) Anti-aliasing via super-resolution and rotation

In order to do that, we generate more than 1 ray from camera at each pixel, then calculate the average value of those rays to be the final value. For rotation, I rotated sample points based on the sample cell center by 26.6 degree.

(5) Texture mapping and normal/displacement texture

We first load texture from disk, then each time we find intersection point, we need to calculate the corresponding u/v value to determine the color of this point based on texture.

In order to get u/v coordinate, we do the following procedure:

$$(u, v) = ((\mathbf{pp_0} \cdot \mathbf{t}, \mathbf{pp_0} \cdot \mathbf{c}) + 1)/2$$

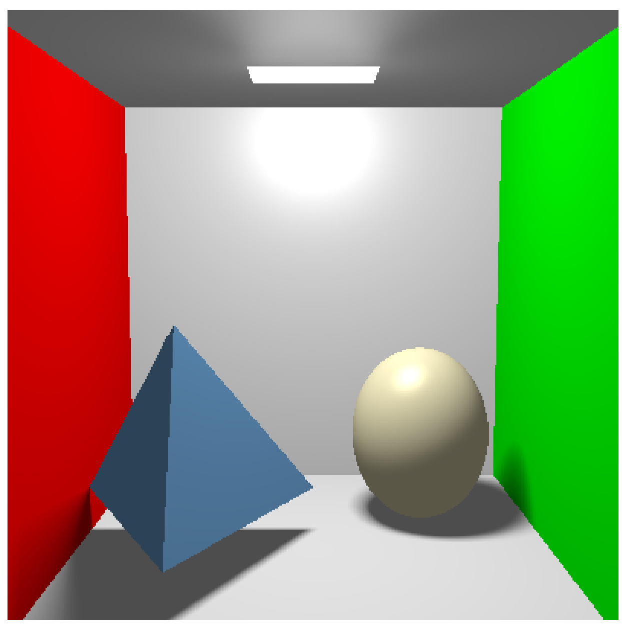
For normal and displacement texture, we use normal texture color as new normal by the following process: assume normal texture value is n, then we need to calculate $\frac{n}{255} \times 2 - 1$ to convert n into range $(0, 1)$, then we transfer n from tangent space into world space with the following matrix

$$\begin{bmatrix} t_x & b_x & n_x & 0 \\ t_y & b_y & n_y & 0 \\ t_z & b_z & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

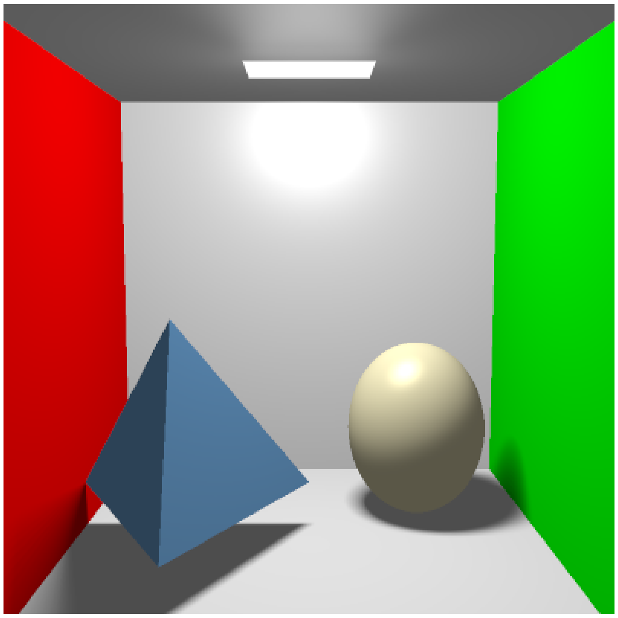
For displacement texture, we use first get value h of displacement texture at position u, v, then we calculate $1 - h$ to be the distortion value, and we translate intersection position along normal direction by $1 - h$ unit.

3 RESULTS

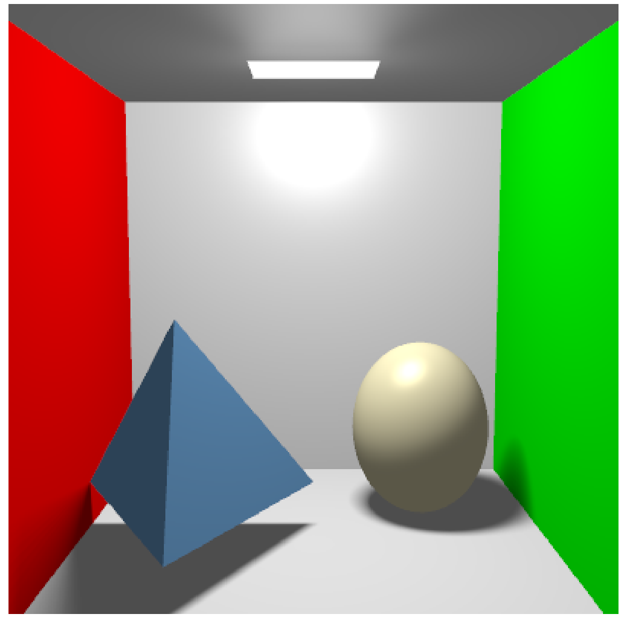
- Cornell-box without anti-aliasing:



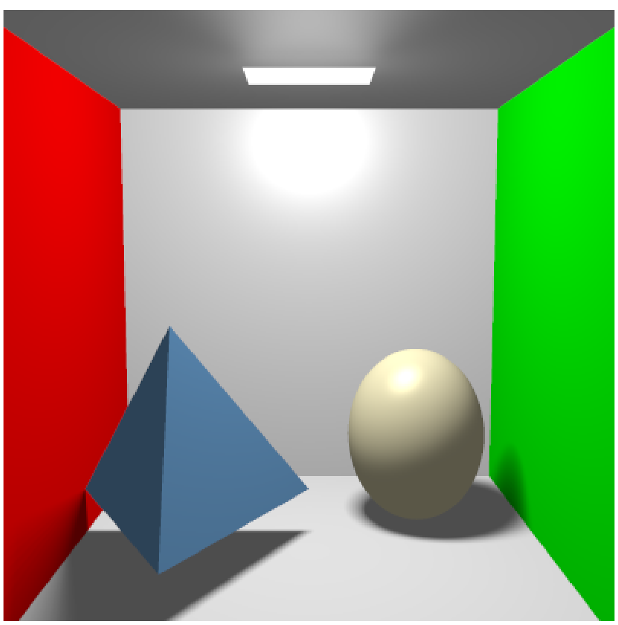
- Cornell-box with super-resolution(4-ray each pixel):



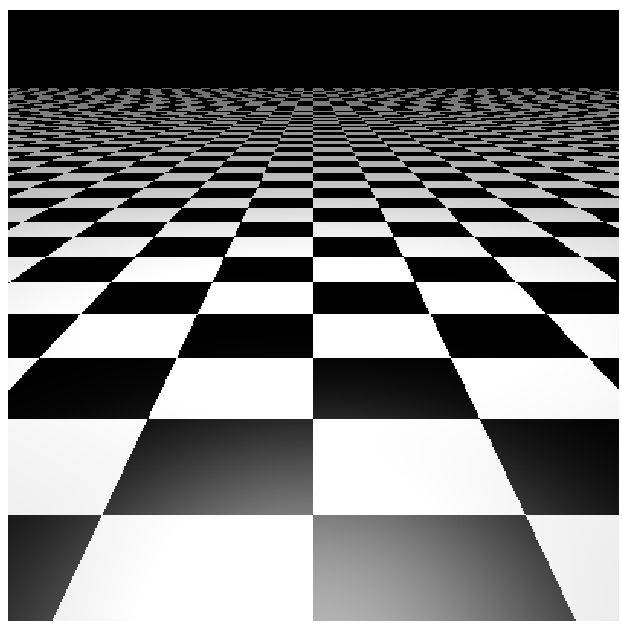
- Cornell-box with super-resolution(16-ray each pixel)



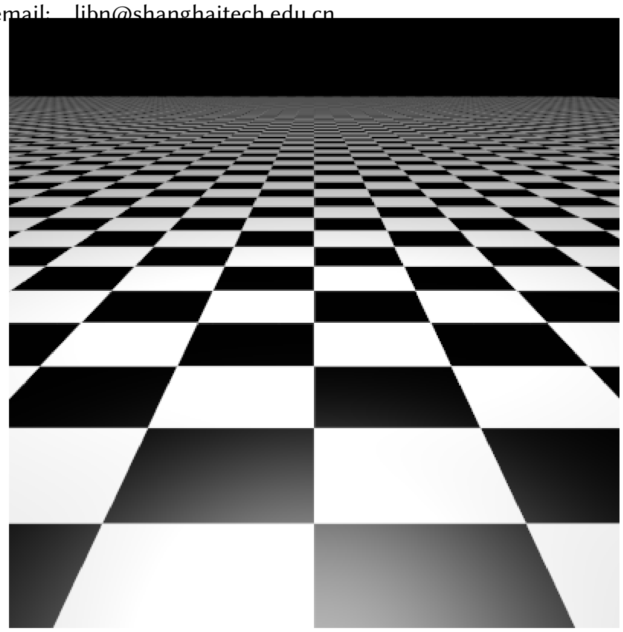
- without anti aliasing



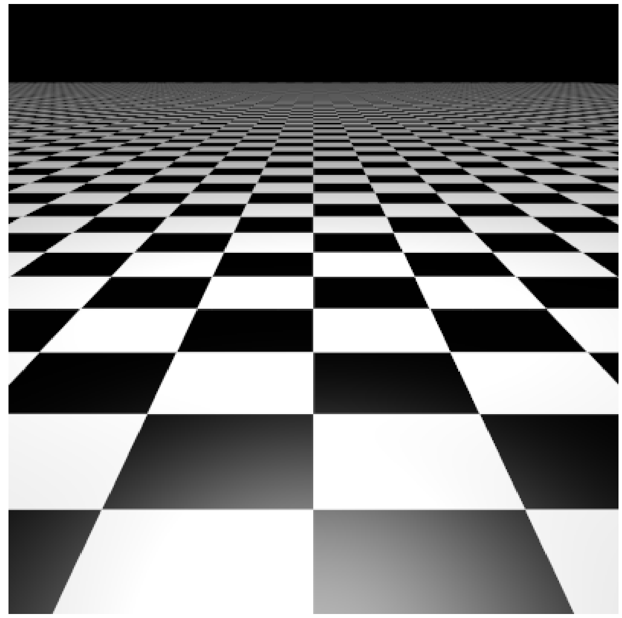
- Cornell-box with rotation anti-aliasing(4-ray each pixel)



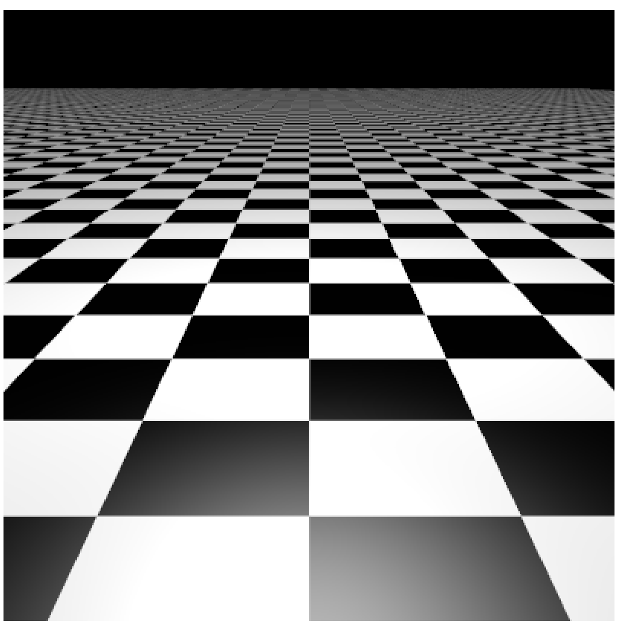
- anti aliasing with super resolution



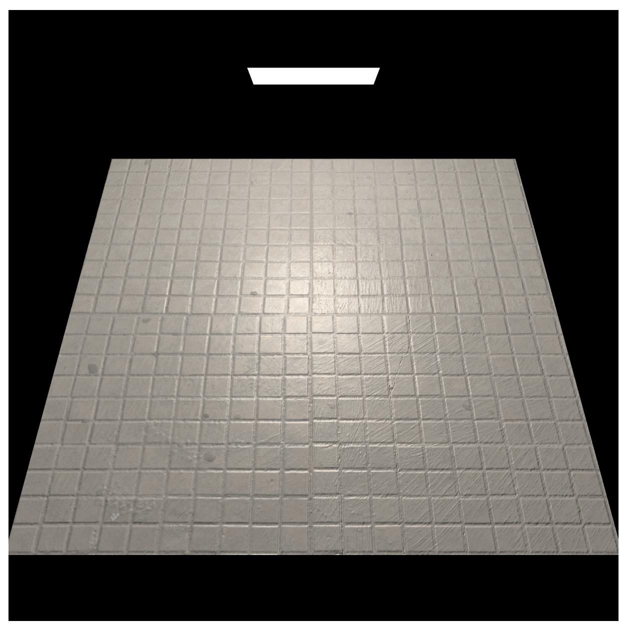
- anti aliasing with rotation



- Apply normal texture



- anti aliasing with rotation and super resolution



- Apply displacement texture

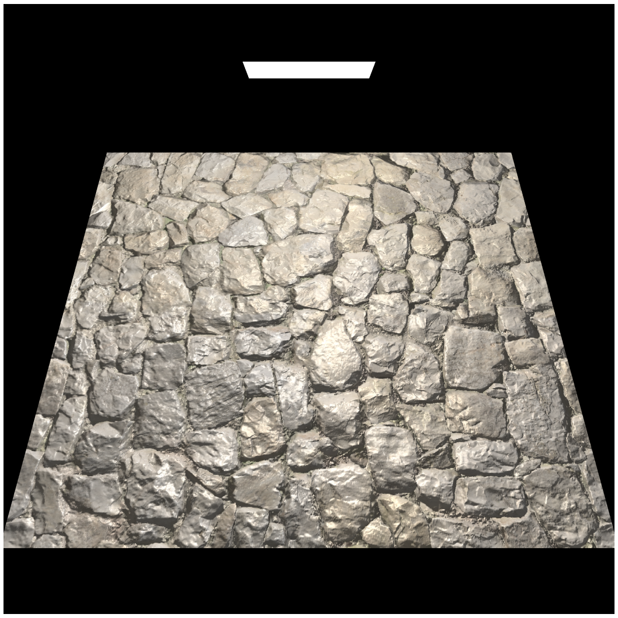


Fig. 1. only normal map

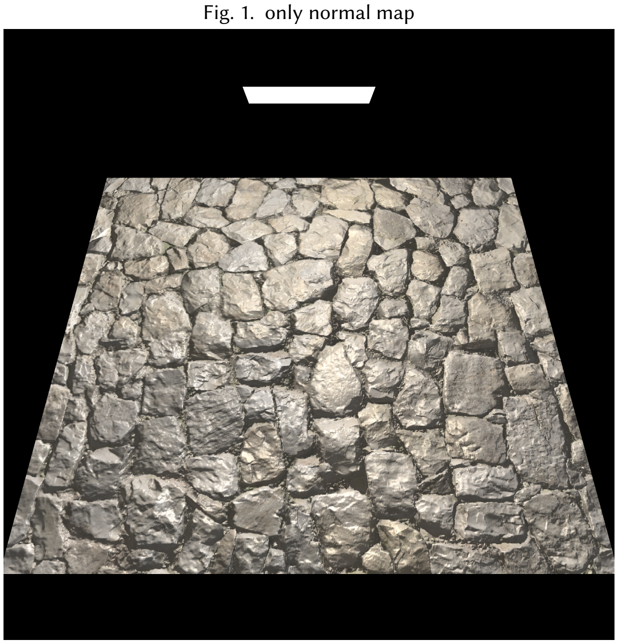


Fig. 2. normal + displacement map