**Interactive Computer Graphics: Midterm Exam**

1. Final Project (5%)

What is your term project for this semester? What are the technical difficulties involved in the project? (You can refer to the project listing).

1. My term project: Image morphing. There are some confusing points to implementing the point warping

2. Visibility (15%)

(a) Painter's Algorithm (5%)

Ba-Zhu (霸主) is one of the most famous internet celebrities who go viral (爆紅) during 2021. He also has many different identities, such as police officer, judge, doctor…, etc. What is the drawing order in the picture below, if we use the painter's algorithm?(The answer should be like 1,2,3,…, 9.)

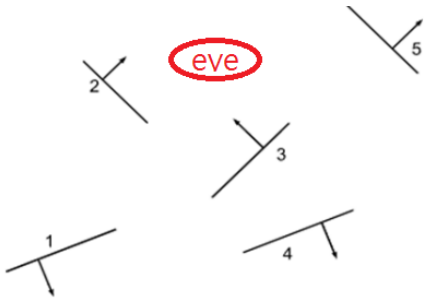

2.

(a).

6, 8, 2, 1, 5, 4, 3, 7, 9

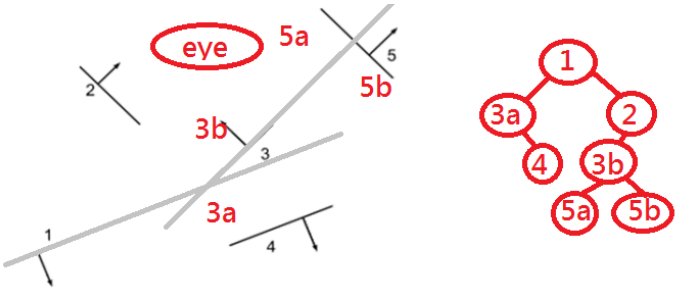
(b) BSP Tree (5%)

BSP tree is an algorithm to address the problem painter's algorithm faces. Construct the BSP tree for the following figure. Use face 1 as the root.




(數字選取需由小到大、且若有需要切分，統一以：左 a 右 b)

(b).



(c) Display Sequence (5%)
What is the display sequence if the eye is placed in the position before faces 3 and 2, but at the back of face 5.

(c). 4, 3a, 1, 2, 5b, 3b, 5a

3 Error Diffusion Dithering (10%)
(a) How do you show the color of rgb(5, 5, 5), when you only have the color of rgb(4, 4, 4) and rgb(8, 8, 8)? Draw your design pattern and explain briefly. (Hint: draw a pattern with at least 4 squares (but no more than 16 squares) with white square represent color rgb(4, 4, 4), and black represent color rgb(8, 8, 8)) (5%)

3.
(a).
取最接近(5, 5, 5) 的 (4, 4, 4) 即可
或者 使用一個4*4的diffusion pattern來處理

| | |
|---|---|
| * | 3/6 |
| 2/6 | 1/6 |

(b) The following is the pseudocode of Floyd-Steinberg error diffusion dithering.

```
for each y from top to bottom
    for each x from left to right
        oldpixel = pixel[x][y]
        newpixel = floor(oldpixel / quant_number) * quant_number
        pixel[x][y] = newpixel
        quant_error = oldpixel - newpixel
        pixel[x+1][y] := pixel[x+1][y] + quant_error * 3/6
        pixel[x][y+1] := pixel[x][y+1] + quant_error * 2/6
        pixel[x+1][y+1] := pixel[x+1][y+1] + quant_error * 1/6
```

Approximate such a matrix using Floyd-Steinberg error diffusion if we have elements of the form 13×N only (i.e. 0, 13, 26, 39, … etc.), give the result of the 2×2 upper left matrix. (5%)

| | | |
|---|---|---|
| 19 | 48 | * |
| 30 | 25 | * |
| * | * | * |

(b).
step1:

| | | | | | |
|---|---|---|---|---|---|
| **13**+6 | 48+**3** | * | -> | **13** | 51 | * |
| 30+**2** | 25+**1** | * | | 32 | 26 | * |
| * | * | * | | * | * | * |

step2:

| | | | | | |
|---|---|---|---|---|---|
| **13** | **39**+12 | *+**6** | -> | **13** | **39** | * |
| 32 | 26+**4** | *+**2** | | 32 | 30 | * |
| * | * | * | | * | * | * |

step3:

| | | | | | |
|---|---|---|---|---|---|
| **13** | **39** | * | -> | **13** | **39** | * |
| **26**+6 | 30+**3** | * | | **26** | 33 | * |
| *+**2** | *+**1** | * | | * | * | * |

step4:

| | | | | | |
|---|---|---|---|---|---|
| **13** | **39** | * | -> | **13** | **39** | * |
| **26** | **26**+**7** | *+* | | **26** | **26** | * |
| * | *+* | *+* | | * | * | * |

4. Transformation Matrix (10 %)
“Attack on Titan (進擊的巨人)”, AoT in short, might be the greatest comic of the 21st century. In addition to the interesting storyline and beloved group of characters, the most fascinating part is: that lots of details, metaphors are embedded in it. For example, the shape and relative position of mainland Merley (馬萊) and Paradis island (帕拉迪島) shown in Fig 4-1, is completely the same as the African continent (非洲大陸) and Madagascar (馬達加斯加島) show in Fig 4
-2, only with horizontally upside down show below.

In AoT, there are several locations such as “Mitras (密特拉教 aka 王政所在地)”, “The Basement (地下室)”, Reiss Chapel (雷斯教堂), “”. Assume their original coordinates are shown below.

| | | | |
|---|---|---|---|
| Mitras | (-1950, 4670) | The Basement | (-2010, 5370) |
| Eren 1st Prison | (-1810, 4790) | Reiss Chapel | (-1830, 4170) |

You should
(1) set “Mitras” to the origin (0, 0).
(2) mirror horizontally.
(3) scale 1/200.
for example, the coordinates of “Eren 1st Prison” become (0.7, -0.6)

(a) What are the new coordinates of “The Basement” and “Reiss Chapel”? (座標請以十進位小數表示之) (5%)
(b) What is the transformation matrix from the real world (以北為上的世界地圖) (old coordinates) to the AoT world map (new coordinates)? (矩陣請以十進位小數表示之) (5%)
4.
(a)
(1) set Mitras to (0, 0), each point should translate (+1950, -4670)
The Basement: (-60, 700)
Eren 1st Prison: (140, 120)
Reiss Chapel: (120, -500)
(2) mirror horizontally, means +y -> -y and -y -> +y
The Basement: (-60, -700)
Eren 1st Prison: (140, -120)
Reiss Chapel: (120, 500)
(3) scale 1/200, means each point should multiply 0.005
The Basement: (-0.3, -3.5)
Eren 1st Prison: (0.7, -0.6)
Reiss Chapel: (0.6, 2.5)
(b)

| | | |
|---|---|---|
| 0.005 | 0 | 0 |
| 0 | 0.005 | 0 |
| 0 | 0 | 1 |

*

| | | |
|---|---|---|
| 1 | 0 | 0 |
| 0 | -1 | 0 |
| 0 | 0 | 1 |

*

| | | |
|---|---|---|
| 1 | 0 | 1950 |
| 0 | 1 | -4670 |
| 0 | 0 | 1 |

=

| | | |
|---|---|---|
| 0.005 | 0 | 9.75 |
| 0 | -0.005 | 23.35 |
| 0 | 0 | 1 |

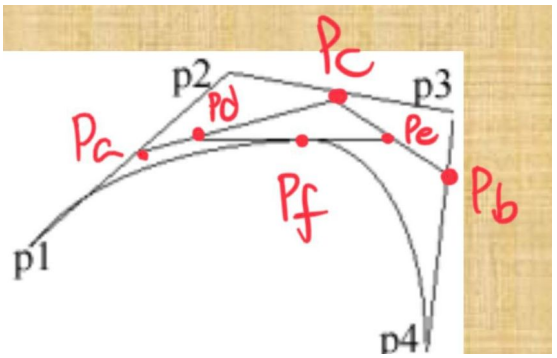
5. Shading (10%)
(a) What are the relationships and differences between vertex shader and fragment shader? (5%)
(b) Briefly describe the three kinds of shading in homework1(Flat shading, Gouraud shading, Phong shading) and compare their differences. (5%)

5.
(a). Vertex Shader: 作用於每個頂點, 通常是處理從世界空間到螢幕座標的座標轉換, 後面緊接的是光柵化(用pixel畫出來)。
Fragment Shader: 作用於每個螢幕上的片元(可近似理解為畫素), 通常是來計算顏色。其中webGL會自動將三頂點的顏色做線性內插。

(b).Flat shading: 又稱平面打光法, 整個面用一個法向量, 該面只有一個顏色

Gouraud shading: 三頂點都有其法向量, 得出各頂點的顏色後, 再用雙線性內插算出整個面的顏色
Phong shading: 三頂點都有其法向量, 先做雙線性內插得出個點的法向量, 再根據該法向量計算顏色

6. Curves and surfaces (10%)
If we have four control points for the Bezier curve. P1(2, 4, 4), P2(4, 8, 6), P3(8, 2, 4), P4(10, 6, 2)
(a) In order to draw this curve, we split it by half. What are the new control points of the two split curves? (Please give exact numbers.) (5%)
(b) To design airplane wings, which kind of curves should be used? Bezier curve or B-splines? Briefly explain your reason. (5%)
6.
(a).
完整算法(標點如右圖):
pa = 1/2 (P1 + P2) = (3, 6, 5)
pb = 1/2 (P3 + P4) = (9, 4, 3)
pc = 1/2 (P2 + P3) = (6, 5, 5)
pd = 1/2 (pa + pc) = 1/2 (9, 11, 10) = (4.5, 5.5, 5)
pe = 1/2 (pb + pc) = 1/2 (15, 9, 8) = (7.5, 4.5, 4)
pf = 1/2 (pd + pe) = 1/2 (12, 10, 9) = (6, 5, 4.5)




new control point:
left: p1, pa, pd, pf
right: pf, pe, pb, p4

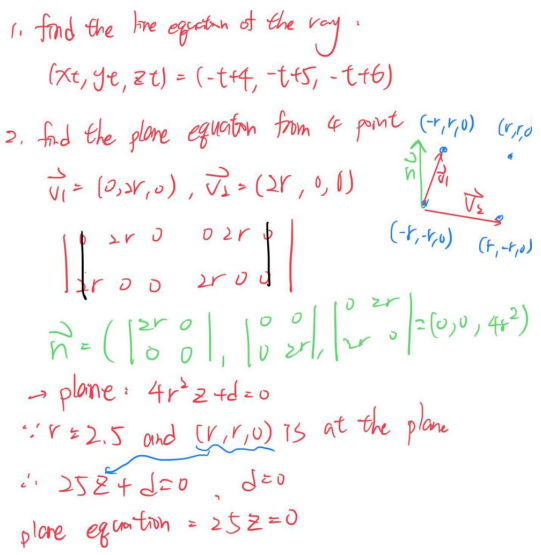
(b). 當然是 B-splines, 因為其保證經過運算後之曲面(or 線) 二次微分也連續。二次微分連續能讓動能更省, 並不會導致而外的干擾

7. Model View Matrix (10%)
In homework 1, Multiplying the model's vertex position by the Model-View Matrix can convert the vertex from model coordinates to world coordinates. Please see the shortcode in homework1 given below. If we exchange the order of translation and rotation, will it affect the result? If so, please describe the difference.

```
// Setup Model-View Matrix
mat4.identity(mvMatrix);
mat4.translate(mvMatrix, [0, 0, -40]);
mat4.rotate(mvMatrix, degToRad(teapotAngle), [0, 1, 0]);
```

7. 若model的中心座標為(0, 0), 則交換兩行代碼, 會使其從自轉變成繞y軸公轉

7_another . Ray objects intersection (8%)
In ray tracing, we need to judge if the ray intersects with a shape or not. Given a ray in 3d (xt, y t, zt) = (vx, vy , vz)×t + (ox, oy , oz ) , t ≥0 , and a square formed by 4 points (-r, -r, 0), (-r, r, 0), (r, r, 0), (r, -r, 0), where (vx, vy , vz ) = (− 1,− 1,− 1) and (ox, oy, oz ) = (4, 5, 6), r = 2.5. Check if the ray intersects with the square, and calculate the intersection point if any one of them exists.




8.

Affine Transform (10%)

We know that vertex normals and face normals are important for a rasterization pipeline. Please show if translation, rotation, scaling, and shearing effects the vertex normal, face normal, and face area or not. (12%)

| | vertex normal | face normal | face area |
|---|---|---|---|
| translation | no | no | no |
| rotation | yes | yes | no |
| scaling | no (不確定) | yes (不確定) | yes |
| shearing | yes | yes | no |

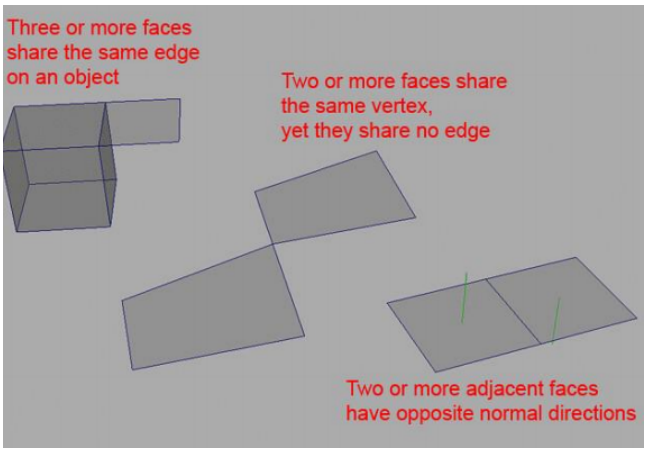
9. Heckbert's Regular Expression (10%)

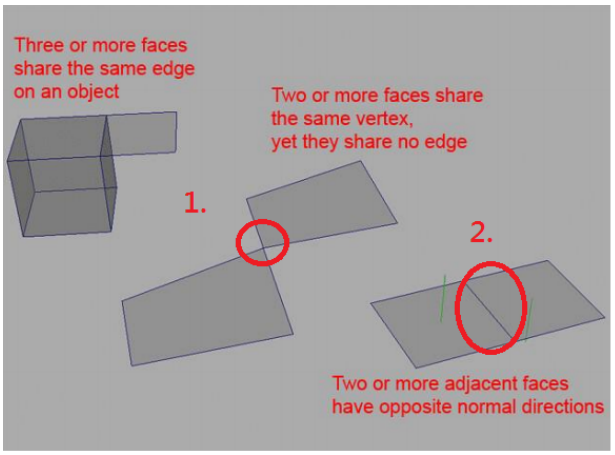
The contributions of light to an image can be considered as light collected over a huge collection of paths through an environment. When thinking about the possible paths, we need to consider both specular and diffuse bounces. If we use a regular expression for the paths considered in ray tracing, we would have: …

9. 不會再考 (open question)

10. Nonmanifold Meshes (10%)

3D meshes are considered problemistic when the mesh has some unwanted traits as shown in the figure. Please state two reasons why these traits are considered displeasing.



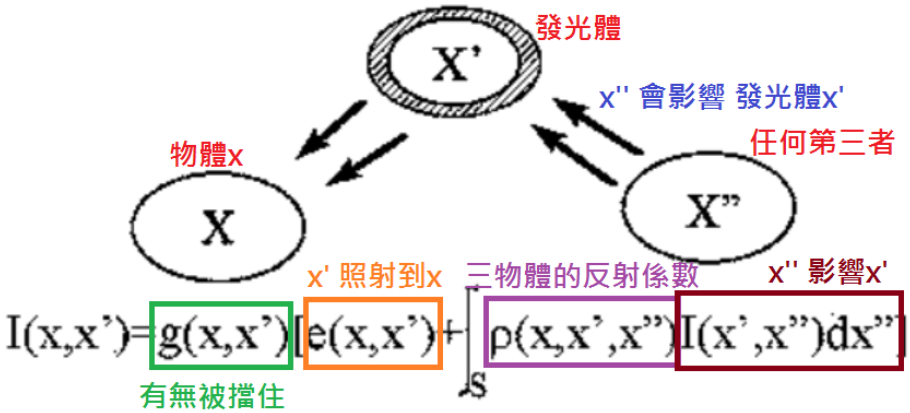




1. we don't know the normal vector of the intersection point
2. we don't know the normal vector of the intersection line

10_another. Ray Tracing (15%)

(a) (5%) What's the weakness of ray-tracing as compared to the Rendering equation? (short answer)

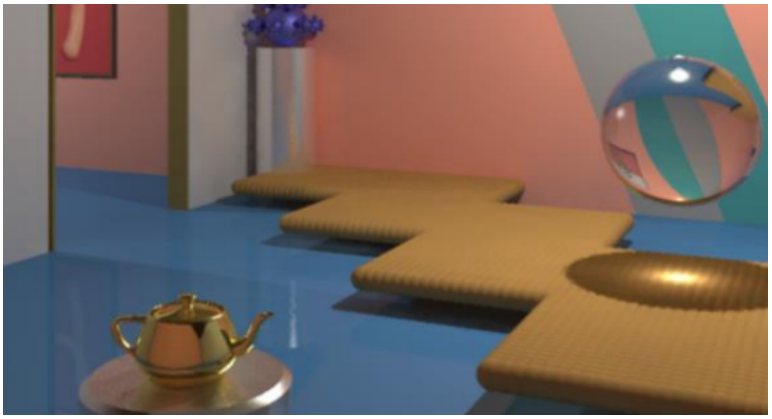
10.




(a).render equation 有考慮到光原本就是能量的方式進行傳遞的問題, 且 也有考慮到物體之間的反射係數 與 形狀係數 (有多少能量會從一個patch轉移到另一個patch), 所以較能真實展示出陰影與光暈的效果

(b) (10%) In the photo below, there are very bright spots under the glass sphere, called caustics. One way to improve the previous ray-tracing is to combine the rays (light packets) shooting from the lights, and store the (location, direction) information at the light-surface intersection points, called photon maps. For caustics, we only store the photon map when rays hit the highly specular surface or pass through a transparent object and finally reach a diffuse surface. After the photon maps are created, we can use ray-tracing to shoot rays from the eye until it hits the surfaces with

the photon maps. Why is this method successful in solving the Rendering Equation? Please give your own algorithm describing the previous solution.



10. b
[ref]: https://www.jianshu.com/p/74a5edef07b1
標準的PM算法由兩個pass組成，分別是Photon Map Generation Pass與Lighting Pass
1. Photon Map Generation Pass: 從光源向各個方向發射photon, 若photon 沒碰撞到場景中的物件就將其projection map的標記設為0。接著以能量的方式紀錄多次跌代的碰種結果於各個pixel 的projection map中。
2. Lighting Pass: 透過以下的光罩公式與單位轉換算出最後的結果。

$$
\begin{aligned}
L_r \quad = \quad & \int_\Omega f_r L_{i,l} \cos\theta_i \, d\omega_i + \quad \text{direct} \\
& \int_\Omega f_{r,s} (L_{i,c} + L_{i,d}) \cos\theta_i \, d\omega_i + \quad \text{specular} \\
& \int_\Omega f_{r,d} L_{i,c} \cos\theta_i \, d\omega_i + \quad \text{caustic} \\
& \int_\Omega f_{r,d} L_{i,d} \cos\theta_i \, d\omega_i \quad \text{indirect}
\end{aligned}
$$