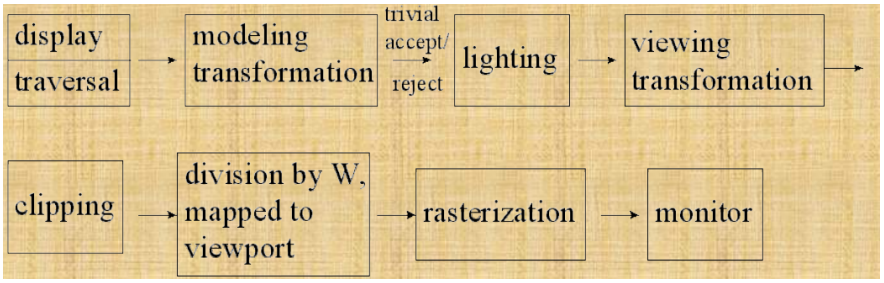
Interactive Computer Graphics Mid-term Exam, 22 Nov. ,2017

1. (15%) According to your homework 1:
(a) (5%) Describe the computer graphics pipeline.
(b) (4%) What is vertex shader and what is fragment shader, what's the relationships and differences between them.
(c) (3%) What's the cons and pros of using indices when drawing a triangle.
(d) (3%) There is a points list [(1,1,1), (0,0,1), (1,0,1)], is it facing the camera at (0, 0, -5) or not? Please explain.

A1.
(a). Describe the computer graphics pipeline.



- display: 逐個展示三角形
- modeling transofrmation: 做平移, 放大, 旋轉, 剪力, …, 等運動
- trivial accpet/reject: 不可能的物體就不去畫出
- lighting: 打光
- viewing transformation: 眼睛位置的運動
- clipping: 計算有一半在視野內, 一半不在的物件
- disvion by W mapped to viewpoint: 做投影透視法
- rasterization: 將結果寫到memory
- monitor: 在螢幕顯示

(b). What is vertex shader and what is fragment shader, what's the relationships and differences between them.

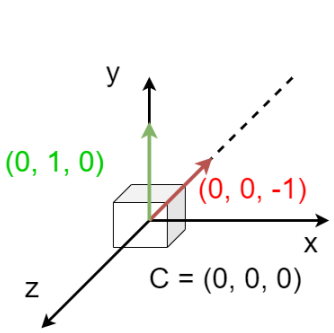
Vertex Shader: 作用於每個頂點, 通常是處理從世界空間到螢幕座標的座標轉換, 後面緊接的是光柵化(用pixel畫出來)。
Fragment Shader: 作用於每個螢幕上的片元(可近似理解為畫素), 通常是來計算顏色。其中webGL會自動將三頂點的顏色做線性內插。

(c). What's the cons and pros of using indices when drawing a triangle.
優點: 若用許多共用的頂點則可以減少頂點的數量。只要有適當的處理(e.g.: 皆於cache時期都被處理), 模型用更少的記憶體資源 且 就能加快渲染的速度
缺點: 若共用點少則沒有其意義。同時, 若沒有經過適當的處理(e.g.: cache miss 很高, 或要多處理的index buffer更為耗時) 的則可能無法加快渲染速度

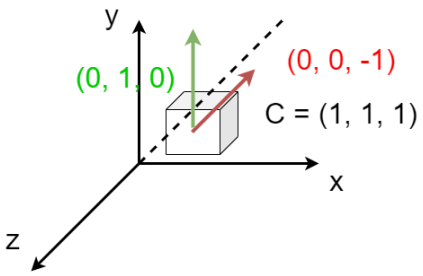
(d). There is a points list [(1,1,1), (0,0,1), (1,0,1)], is it facing the camera at (0, 0, -5) or not? Please explain.
No, 沒有給camera所朝向的方向(view reference point, VRP)

2. (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).
My term project: Image morphing. There are some confusing points to implementing the point warping

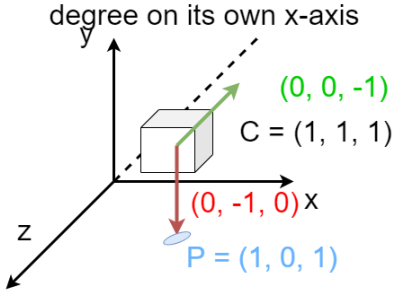
3. (8 %) Suppose the camera is at the (0,0,0) pointing at (0, 0, -1), the up vector is (0, 1, 0). Find the 4x4 transformation that transforms the camera to the position (1, 1, 1), pointing at (1,0,1), and setting the up vector to be (0, 0, -1)



1. move to (1, 1, 1)



2. clockwise rotate 90 degree on its own x-axis



(0, -1, 0) = P - C = (1, 0, 1) - (1,1,1)

1. translate to (1, 1, 1):

Matrix1 =

1	0	0	1
0	1	0	1
0	0	1	1
0	0	0	1

2. clockwise rotate 90 degree on its own x, which is equal to anticlockwise rotate 270 degree on it own x:

Matrix2 =

1	0	0	0
0	cos(270°)	-sin(270°)	0
0	sin(270°)	cos(270°)	0
0	0	0	1

=

1	0	0	0
0	0	1	0
0	-1	0	0
0	0	0	1

Ans: Becasue we need to anticlockwise rotate 270 degree on it own x, we need to transform matrix2 at first. Then, doing the translation by matrix1. Therefore, the answer is Matrix1 * Matrix2:

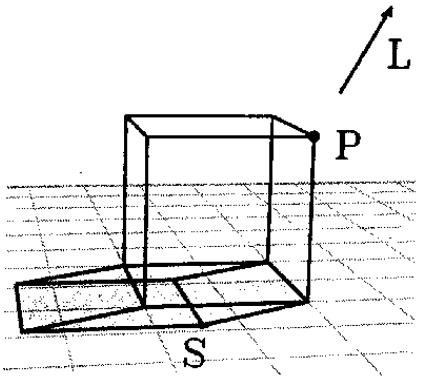
Matirx1 * Matrix2 =

1	0	0	1
0	0	1	1
0	-1	0	1
0	0	0	1

4. (6 %) How to render fake shadows using modified homework #1 Phong shading. (For example, suppose there is only a cube on the floor, and the light is on the upside of the cube)

[Ref]: https://web.cs.wpi.edu/~matt/courses/cs563/talks/shadow/shadow.html

若實作ray tracing 或 radiosity 於HW1則此題可解, 且通常webGL有api可以呼叫



若使用推導算法上則是通過計算shadow矩陣, shadow的點位置可通過該矩陣計算而得
其中又分成以下兩種方式: 1) 無限光源, 2) 區域光源

1) 無限光源:
   以右圖來說, 假設L唯一向量, P為頂點座標, S為頂點座標
   則 S = P - alpha * L
   又我們知道 Zs = 0 (陰影須在z座標為 0)
   則 0 = Zp - alpha * ZL -> alpha = Zp / ZL
   => Xs = Xp - Zp/ZL * XL
   進而推得陰影矩陣為:

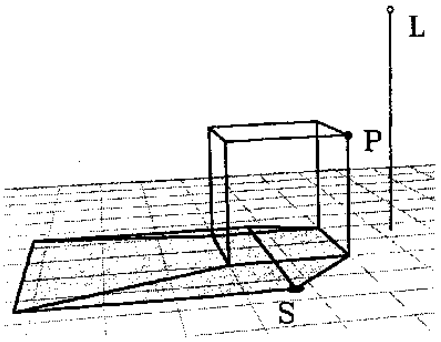
$$M_{shadow} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -x_l/z_l & -y_l/z_l & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

   S = P * Mshadow

2) 區域光源類似解法
   S = P + alpha * (L - P)
   alpha = -Zp / (Zp - ZL)
   Xs = (XLZp - Xp*ZL) / Zp -ZL

$$M_{shadow\_prime} = \begin{vmatrix} -z_l & 0 & 0 & 0 \\ 0 & -z_l & 0 & 0 \\ x_l & y_l & 0 & 0 \\ 0 & 0 & 0 & -z_l \end{vmatrix}$$

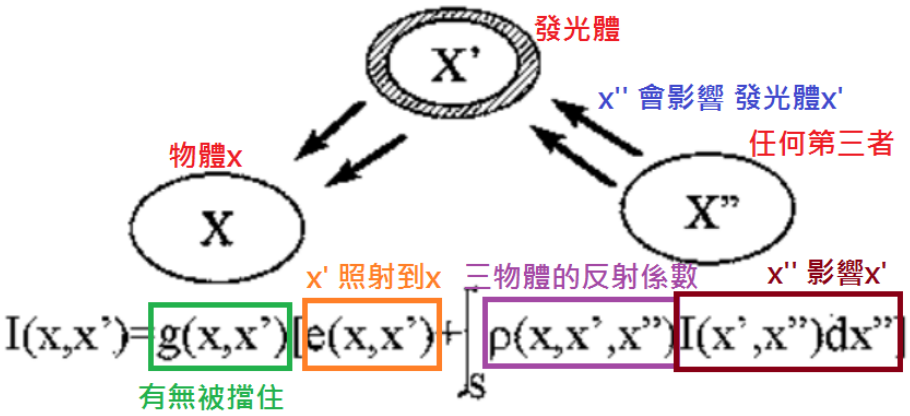
   S = P * Mshadow_prime



5. (12%) Remember that in our course, our professor once shot a laser beam through his ear as an example. He wanted to show that light can penetrate our flesh.
(a) (4%) Can Whitted's recursive ray tracing algorithm handle such effects? Why?
(b) (8%) Refer to the rendering equation. Given the human flesh, considering its complex structures, how to model it correctly, and please explain your method.

5.
(a). Yes, because our flesh is a translucent (半透明) object, so the ray can penetrate it.

(b). 首先, 先寫下render equation, 其中公式關係如下:



而用電腦解決此方程式時, 我們通常會以矩陣表示法來解決, 可以表示成:
BiAi (總能量) = EiAi (若自己為發光體的能量) + Pi * Sumj (Bj Fji Ai) (所有第三者影響Ai的能量)
再經由能量平衡原則, 可以簡化成以下:
Bi = Ei + Pi Sumj (Bj * Fi-j)
其中, 在解決此問題時, 最困難點之一在於要構成flesh 的三角形眾多(導致矩陣過大), 且需要計算好每一個形狀係數Fji 的值
我們可以嘗試使用Chen研究員所開發的逼近法來達到此事, 並優化運算速度
其中的解法名稱稱為, Hemi-cube method, 大致上有四個步驟

1. j投影到i上
2. 算出投影後所佔據的單位比例
3. 算出投影與cube法向量的夾角
4. 計算出Fji的值

接著優化速度的部分, 則由能量大的先向外擴算並計算(此時會將自己歸0), 經過一定的跌代後會漸漸收斂(約100次左右跌代即可), 此時大大的加快的運算速度

6. (15%) A normal vector $\hat{n}$ is defined to be a unit vector that is perpendicular to a given surface. For example, in the three-dimensional Euclidean space, the normal vector of a triangle ABC defined by three points A, B and C can be obtained with $(\overrightarrow{AB} \times \overrightarrow{AC})/||\overrightarrow{AB} \times \overrightarrow{AC}||$. Please answer the following questions:

(a) (10 %)

Given a normal vector $\hat{n} = [n_x, n_y, n_z]^T$ and a transformation matrix M containing some translation, rotation, and uniform scaling effects:

$$M = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

How would you obtain the normal vector $\hat{n}'$ which is the result of transforming $\hat{n}$ by M? Express your solution in mathematical form.

(b) (5 %)

Tom, a naughty boy, likes to see 3-D objects deform in some strange ways. He now adds a non-uniform scaling effect to the transformation matrix M. Is your answer to (a) still correct? If not, briefly explain why.

6.

(a).

[ref]: https://zhuanlan.zhihu.com/p/405547033

assume vector v(x, y, z) is at the plane (built from triangle ABC), and n is the transpose of n^

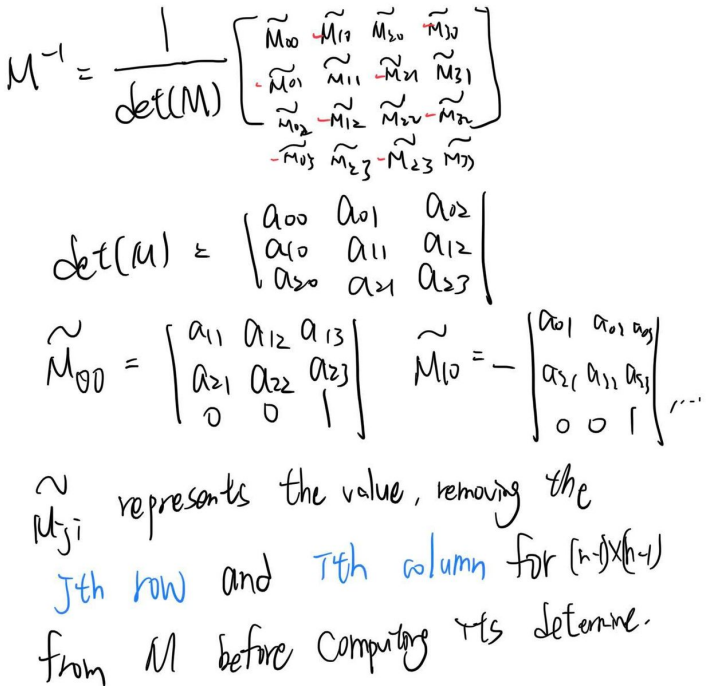
therefore, $n^T \cdot v = 0 = n^T \cdot I \cdot v = n^T \cdot M^{-1} \cdot M \cdot v = 0$

-> $(M^{-T} n)^T (M \cdot v) = 0$

and, $n^{\wedge\prime} = M^{-T} n$. The detail of the M^{-1} is at below:

[ref]:

https://ccjou.wordpress.com/2012/10/04/%E4%B8%89%E9%9A%8E%E9%80%86%E7%9F%A9%E9%99%A3%E5%85%AC%E5%BC%8F/



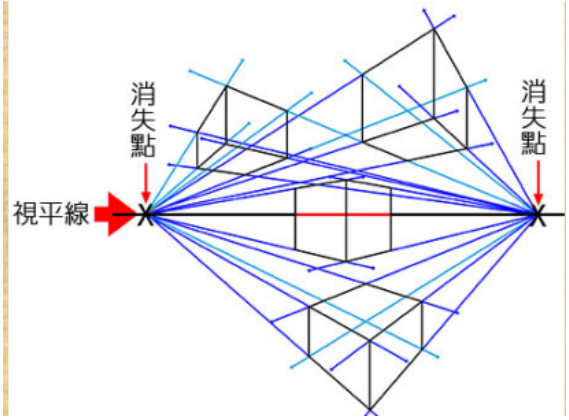
(b). the non-uniform scaling still works for this situation. Because transformation M^{-T} contains all of the transformation cases.

7. (10%)
(a) Please describe briefly about the types of perspective projection using vanishing points. (3%)
(b) The following picture is a part of the 清明上河圖, a famous Chinese painting. Would you please discuss about the projection method the painter 張擇(北宋) has used. What are the advantages about this method? Which part is difficult in computing according to your proposed method? (7%)

7.

(a). 簡單來說, perspective projection 指的就是消失點的投影
以單眼透視法來說, 就是所有平行線消失在一個點上(如右圖)

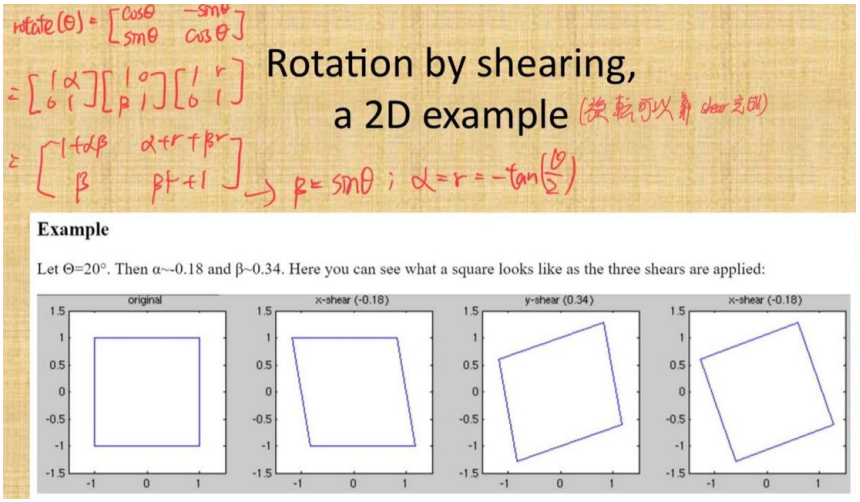
其中Two-point perspective則是消失在兩點上(如下圖)
目前的相機則是採用3點透視法



(b). 清明上河圖是採用移動點透視法(Moving Point Perspective) , 類似走一段路將所看到的景象繪製完後, 再走一段路再繪製, 直到完成繪畫
優點: 能將廣闊的美景收錄在一張紙上, 在沒有計算機的年代是很了不起的藝術作品之一。
此做法有諸多細節, 廣度方面是透過平行斜視法繪製, 且要能保證斜線不能往消失點集中, 否則將無法表示出無限空間的感覺。
深度與高度方面是透過上下關係法, 重疊法所構築, 要確保空間尚能完美融合

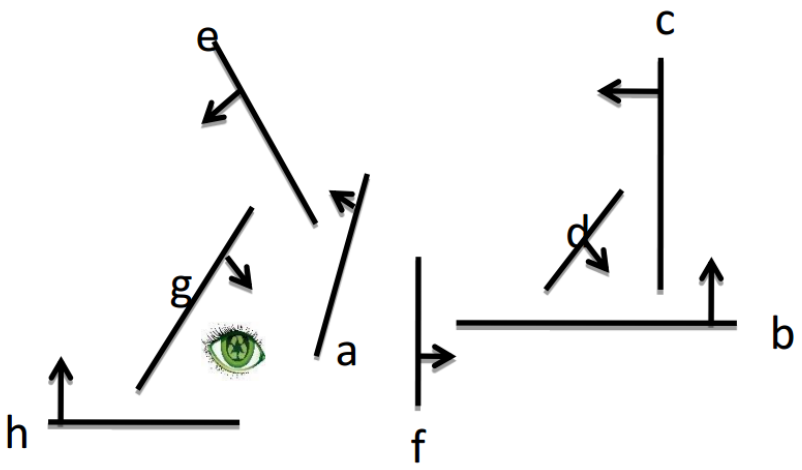
8. (10%) In 2D, a rotation transformation by angle can be specified as a series of shear transformation matrices. Give these matrices, or if it can't be done, prove it.

[ref]: https://www.ocf.berkeley.edu/~fricke/projects/israel/paeth/rotation_by_shearing.html



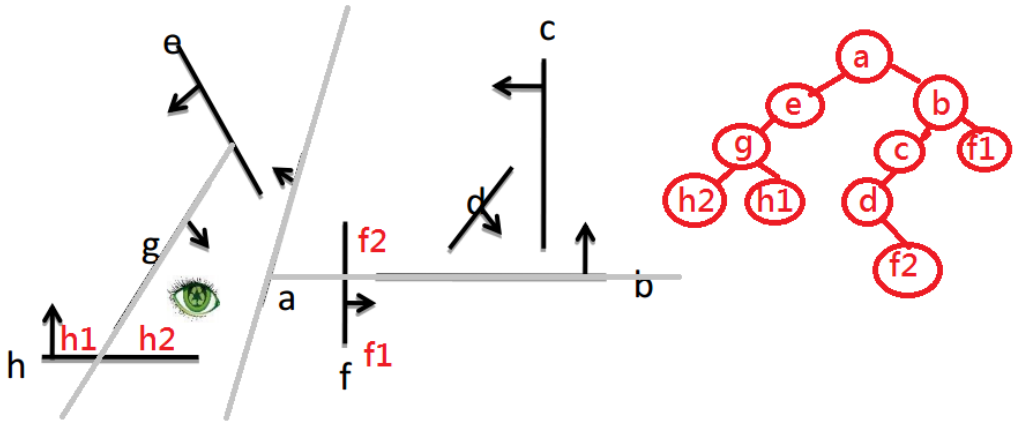
9. BSP Tree (15%)

A. Construct the Binary Space Partitioning (BSP) tree of the model in Fig.1below. Please use node "a" as the root. Split any line segment as you wish, and mark them as b.1, b.2, b.3, b.4 etc. Please choose smaller numbers/alphabets as the sub-tree root node.



B. From the BSP tree in (a), derive the display sequence in terms of the given viewing position in this figure.

A. 如下圖



B. 根據BPS演算法如下:
c, d, f2, b, f1, a, e, h1, g, h2

10 Curves and surfaces. (9%)

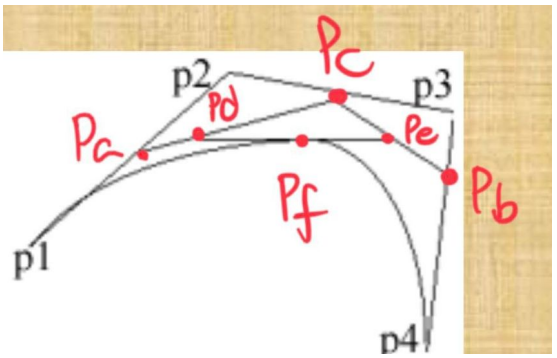
If we have four control points for the Bezier curve.P1 (1, 2, 4), P2 (3, 8, 6), P3 (6, 6, 4), P4 (8, 2,2).

(a) (6%) 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).

(b) (3%) To design airplane wings, which kind of curve should be used? Bezier curve or B-splines? Why? (please give a short answer).

10.
(a).
完整算法(標點如右圖):
pa = 1/2 (P1 + P2) = (2, 5, 5)
pb = 1/2 (P3 + P4) = (7, 4, 3)
pc = 1/2 (P2 + P3) = (4.5, 7, 5)
pd = 1/2 (pa + pc) = 1/2 (6.5, 12, 10) = (3.25, 6, 5)
pe = 1/2 (pb + pc) = 1/2 (11.5, 11, 8) = (5.75, 5.5, 4)
pf = 1/2 (pd + pe) = 1/2 (9, 11.5, 9) = (4.5, 5.75, 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

104. 2.(10%) When using the Cohen-Sutherland line clipping algorithm, how do we check the outcodes to see if a line can be trivially accepted or rejected? (the below answer is from ppt)

Ans: Each coordinate of a line is assigned a 4-bit outcode according to its location relative to the left, right, top, and bottom screen edges.

- First bit set to 0 if: ycoord < yscreen-min
- Second bit set to 0 if: ycoord > yscreen-max
- Third bit set to 0 if: xcoord < xscreen-min
- Fourth bit set to 0 if: xcoord > xscreen-max

If both of a line's outcodes are 0000 (logical OR), trivially accept.
If a logical AND of the lines outcodes is not zero, trivially reject.

97. 5.(15%)In two dimensions, we can specify a line by the equation y = mx + b.
(a)(10%)Find an affine transformation to reflect two-dimensional points about this line.
(b)(5%)Extend your results to reflection about a plane in three dimensions.

In two dimensions, we can specify a line by the equation $y = mx + b$.

1. Find an affine transformation to reflect two-dimensional points about this line.

   This transformation can be constructed in stages. If we do a translation, T, by $-b\hat{y}$ we convert the problem to reflection about a line passing through the origin; the translation matrix is

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix}.$$

   From the slope, $m = \tan\theta$, we can find a rotation so the line is aligned with the x (or y) axis:

$$R = R(-\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

   Now apply a reflection, S, about the x (or y) axis,

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

   Finally we undo the rotation and translation so the sequence is of the form

$$T^{-1}R^{-1}SRT = \begin{bmatrix} \cos 2\theta & \sin 2\theta & -b\sin 2\theta \\ \sin 2\theta & -\cos 2\theta & b\cos 2\theta \\ 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

   For verification purposes, notice that when $b = m = 0$, the matrix correctly simplifies to S.

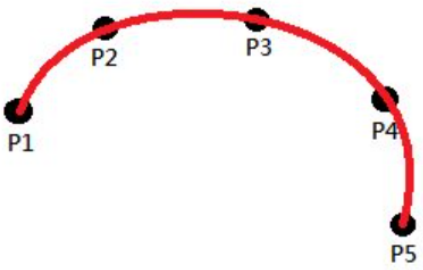
103. 1. (5%) Suppose that we render Bezier patches by adaptive subdivision so that each patch can be subdivided a different number of times. Do we maintain geometric continuity along the edges of the patches? Draw a picture and explain your answer.

Ans: If we have two patches that share an edge and subdivide only the patch on one side of this edge, we can create a crack. The middle shared endpoint on the subdivided patch does not have to lie on the original edge. We can either create an extra triangle from the original endpoints and this middle point to fill the crack or we can triangulate the unsubdivided patch to meet the new divided edge, i.e. we replace the unsubdivided patch with a set of triangles that use three of the edges of the patch and the subdivided edge.

6. (10%) Curves and Surfaces

(1) (5%) Please give a brief comparison (advantages) between the Bezier curve and B-spline.

(2) (5%) If we have control points, P1, P2, …, P5, how to draw a curve through start point P1 and endpoint P5 using B-spline (that is let the curve pass through P1 and P5)? Write down the shortest control points sets to draw. (just like (P1, P2, P2, P3, P4, P5))



Ans: (2) (P1, P1, P1, P1, P2, P3, P4, P5, P5, P5, P5, P5)