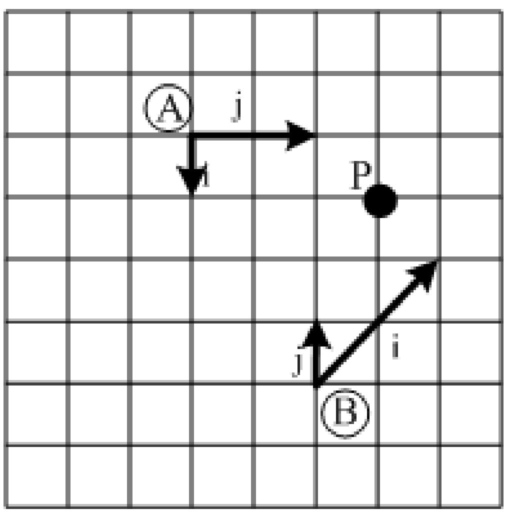
CPSC 314 - Quiz 1 Solution

October 26, 2004

1. Transformation as a Change of Coordinate Frame



(a) (7 points) Specify the coordinates of point P with respect to coordinate frames A and B.

$$p_a = \begin{bmatrix} 1 & 1.5 \end{bmatrix}$$
$$p_b = \begin{bmatrix} 0.5 & 2 \end{bmatrix}$$

(b) (8 points) Derive a transformation that takes a point from frame A to frame B, i.e., determine $M_{A\rightarrow B}$, where $P_B = P_A M_{A\rightarrow B}$. Verify your solution using your answer to part (a).

First the translation between the origins, in row 3. Next, i_a in terms of i_b, j_b in row 1 and j_a in terms of i_b, j_b in row 2.

$$M_{A\rightarrow B} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & -2 & 0 \\ -1 & 6 & 1 \end{bmatrix}$$

2. (a) (7 points) What will the following transformation in 3D homogeneous coordinates do to a unit cube centered at the origin (use row vectors)? Sketch your answer.

$$M = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 2 & 5 & 0 & 1 \end{pmatrix}$$

M=ST= $\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2 & 5 & 0 & 1 \end{pmatrix}$ Points are first scaled, then translated.

The cube is first stretched into a brick (2x1x3), and then translated 2 units along the X axis, and 5 units along the Y axis.

(b) (7 points) What is the inverse of this transformation matrix (3D homogeneous coordinates)? Explain your computation.

$M^{-1} = (ST)^{-1} = T^{-1}S^{-1}$

$T^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\mathbf{2} & -5 & 0 & 1 \end{pmatrix}$ $S^{-1} = \begin{pmatrix} 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

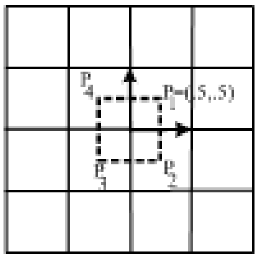
$M^{-1} = \begin{pmatrix} 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/3 & 0 \\ -1 & -5 & 0 & 1 \end{pmatrix}$

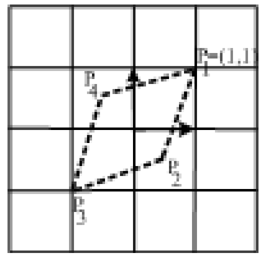
(c) (6 points) Give the sequence of OpenGL transformations that would produce the same transformation matrix (row vectors) as in (a) above.

```
glTranslatef(2, 5, 0);
glScalef(2, 1, 3);
```

3. Transformations

   (a) (15 points) Describe the 2D transformation matrix that shears the unit square centered at the origin as shown in the figure below. Show your work.





Since the center of the square does not move it is possible to solve in 2D without homogeneous coordinates (2x2 systems).

Option 1 (using row vectors)

Rotate -45 degrees, scale along X axis by 2, rotate back

$$\begin{pmatrix} cos(-\pi/4) & sin(-\pi/4) \\ -sin(-\pi/4) & cos(-\pi/4) \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} cos(\pi/4) & sin(\pi/4) \\ -sin(\pi/4) & cos(\pi/4) \end{pmatrix} = \begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}$$

Option 2 (using row vectors)

Define 2x2 transformation matrix $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$

Then solve $\begin{pmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{pmatrix} M = \begin{pmatrix} 1 & 1 \\ 0.5 & -0.5 \end{pmatrix}$

$$M = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 1 \\ 0.5 & -0.5 \end{pmatrix} = \begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}$$

   (b) Answer yes/no (no explanation). All the transformations are in 3D.

      i. (5 points) Does rotation preserve lengths? yes, a rotation matrix is orthonormal.

      ii. (5 points) Is scale1 * scale2 = scale2 * scale1? yes, for S_1, S_2 diagonal matrices $S_1S_2 = S_2S_1$

      iii. (5 points) Is translate * scale = scale * translate? no, see question 2.

4. (a) (5 points) Given 2 points $P_1 = (1, 0)$ and $P_2 = (3, 3)$ in 2D compute the implicit equation $Ax + By + D = 0$ of the line passing through those points. Show your work.

   The line is given by $dy \cdot x - dx \cdot y + x_1 y_0 - x_0 y_1$

   Therefore $A = dy = 3$ $B = -dx = -2$ $D = x_1 y_0 - x_0 y_1 = -3$

   (b) (5 points) Does the point $P = (2, 2.5)$ lie on the segment P_1, P_2? Explain your computation.

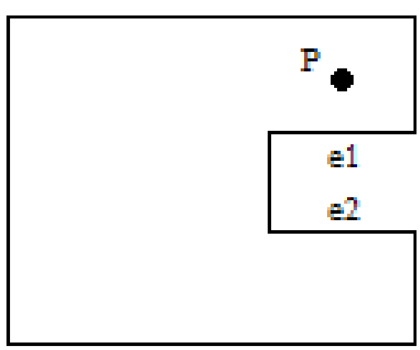
   $3 \cdot 2 - 2 \cdot 2.5 - 3 \neq 0 \rightarrow$ Point not on line

5. (a) (8 points) Given a convex planar polygon $P = P_1, P_2, \ldots, P_n$, in 2D and a point P write an algorithm (pseudo code) to test if P is inside/outside the polygon.

```
For i = 1..n
  Find line l(x,y):=implicit equation for line from P(i) to P(i+1 mod n)
  D(i) = l(p)

If all Di have same sign,
  return "p is inside"
else
  return "p is outside"
```

   (b) (3 points) Will your algorithm work for simple non-convex polygons? If yes, explain. If not, give a counter-example.

   No. Point P will not be on the same side of e_1 and e_2 traversing clock-wise or counter clock wise.



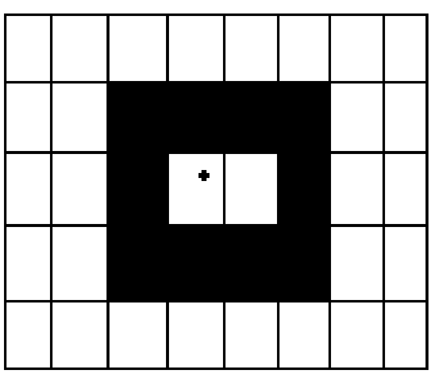
   (c) (4 points) Write an algorithm (pseudo code) for determining whether a given polygon is a simple, convex polygon.

```
For i = 1..n
  If interior angle at Pi is greater than 180 degrees
    return false

For j = 1..n
  For k = j+1..n
    if edges j, k intersect return false

return true
```

6. (10 points) Will the following variation of flood-fill paint the polygon correctly? Answer yes/no and explain.

```
FloodFill(Polygon P, int x, int y, Color C)
if not(OnBoundary(x,y,P) or Colored(x,y,C))
begin
   FloodFill(P,x+1,y,C);
   FloodFill(P,x-1,y,C);
   FloodFill(P,x,y+1,C);
   FloodFill(P,x,y-1,C);
   PlotPixel(x,y,C);
end
```

No. It goes into infinite recursion for ANY polygon larger than one pixel, e.g.:



Start on left, move right,

then left, then right, then left....