

# CPSC 314

## Assignment 1

Due 4pm, Friday October, 14 2004  
(handin box in the CICSR basement)

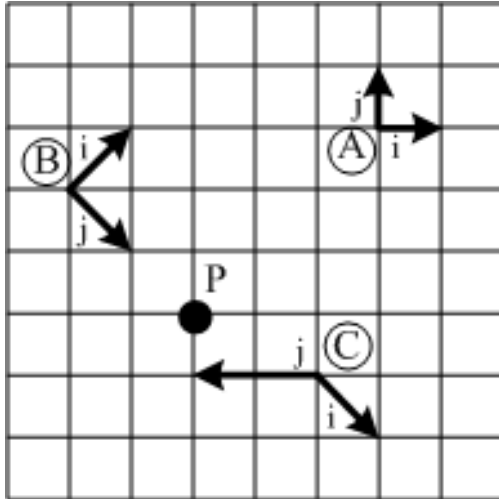
Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, continue on the back of the page.

Name: \_\_\_\_\_

Student Number: \_\_\_\_\_

Theory	/ 20
Question 1	/ 3
Question 2	/ 3
Question 3	/ 4
Question 4	/ 3
Question 5	/ 4
Question 6	/ 3
Question 7 (bonus)	/ 4
Programming Assignment	/ 80
TOTAL	/ 100

1. (3 points) Transformation as a Change of Coordinate Frame



Derive a transformation that takes a point from frame  $C$  to frame  $B$ , i.e., determine  $M_{C \rightarrow B}$ , where  $P_B = M_{C \rightarrow B} P_C$ . Verify your solution using the coordinates of  $P$  with respect to the different frames (see your answers in assignment 0).

We must first express the vector  $C_i$  in terms of  $B_i$  and  $B_j$ . You can solve this visually or solve for  $a$  and  $b$  in the equation

$$aB_i + bB_j = C_i$$

$$\begin{pmatrix} B_{ix} & B_{jx} \\ B_{iy} & B_{jy} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} C_{ix} \\ C_{iy} \end{pmatrix}$$

Then we must express the vector  $C_j$  in terms of  $B_i$  and  $B_j$ . Again, you can solve this visually or solve for  $c$  and  $d$  in the equation

$$cB_i + dB_j = C_j$$

$$\begin{pmatrix} B_{ix} & B_{jx} \\ B_{iy} & B_{jy} \end{pmatrix} \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} C_{jx} \\ C_{jy} \end{pmatrix}$$

Now we must express the vector representing the origin translation between  $B$  and  $C$  in terms of  $B_i$  and  $B_j$ . You can solve this visually or solve for  $e$  and  $f$  in the equation

$$eB_i + fB_j = C_O - B_O$$

$$\begin{pmatrix} B_{ix} & B_{jx} \\ B_{iy} & B_{jy} \end{pmatrix} \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} C_{Ox} - B_{Ox} \\ C_{Oy} - B_{Oy} \end{pmatrix}$$

For convenience we can combine the previous three matrix equations into one large matrix equation:

$$\begin{pmatrix} B_{ix} & B_{jx} \\ B_{iy} & B_{jy} \end{pmatrix} \begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix} = \begin{pmatrix} C_{ix} & C_{jx} & C_{Ox} - B_{Ox} \\ C_{iy} & C_{jy} & C_{Oy} - B_{Oy} \end{pmatrix}$$

$$\begin{aligned} \begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix} &= \begin{pmatrix} B_{ix} & B_{jx} \\ B_{iy} & B_{jy} \end{pmatrix}^{-1} \begin{pmatrix} C_{ix} & C_{jx} & C_{Ox} - B_{Ox} \\ C_{iy} & C_{jy} & C_{Oy} - B_{Oy} \end{pmatrix} \\ \begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix} &= \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & -2 & 4 \\ -1 & 0 & -3 \end{pmatrix} \\ \begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix} &= \begin{pmatrix} 0 & -1 & 0.5 \\ 1 & -1 & 3.5 \end{pmatrix} \end{aligned}$$

Finally, we need to turn the matrix into a homogenous transformation matrix:

$$\begin{pmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0.5 \\ 1 & -1 & 3.5 \\ 0 & 0 & 1 \end{pmatrix}$$

Which is the matrix  $M_{C \rightarrow B}$ .

- (3 points) Given a line segment  $S = (P_0, P_1)$  in 2D and a point  $P$ , write an algorithm to find if the point is on the line segment. Note that your algorithm should operate in the Euclidean (not discrete) space.

From the parametric equation for  $S$

$$l(t) = P_0 + t(P_1 - P_0)$$

$$t \in [0, 1]$$

We know that if  $P$  is on  $S$ , then the vector from  $P_0$  to  $P$  is a linear combination of the vector from  $P_0$  to  $P_1$  whose coefficient is bounded by 0 and 1. To verify this we define

$$\lambda_x = \frac{P_x - P_{0x}}{P_{1x} - P_{0x}}$$

and

$$\lambda_y = \frac{P_y - P_{0y}}{P_{1y} - P_{0y}}$$

If  $P$  is on  $S$ , then  $\lambda_x = \lambda_y$  and  $\lambda_x \in [0, 1]$ . This is because  $\lambda_x = \lambda_y$  implies that the vector  $P - P_0$  is a scalar multiple of  $P_1 - P_0$ . Note that there is a possibility of division by zero with this method, so the denominators must be checked first. For instance, if  $P_{1x} - P_{0x} = 0$ , then  $P$  is on the line only if  $P_x = P_{0x}$  and if  $0 \leq \frac{P_y - P_{0y}}{P_{1y} - P_{0y}} \leq 1$ .

An alternative method uses the knowledge that for any point on a line segment, the sum of the two distances between the point and the endpoints equals the distance between the endpoints (draw a picture to understand this). So if  $d(P, P_0) + d(P, P_1) = d(P_0, P_1)$  where  $d$  is a function that returns the Euclidean distance of two points, then the point is on the line segment.

- (4 points) Decompose the following complex transformations in homogeneous coordinates into a product of simple transformations (scaling, rotation, translation, shear). Pay attention to the order of transformations.

(a) (1 point)

$$\begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix} = R * S$$

When operating in the global coordinate space, it is a uniform scaling  $S$  followed by a rotation  $R$ .  $R$  is a 90 degree rotation around the  $Z$  axis, and  $S$  scales all axes by  $\frac{1}{2}$ .

(b) (1 point)

$$\begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 2 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = T * S$$

When operating in the global coordinate space, it is a scaling  $S$  followed by a translation  $T$  where

$$T = \begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(c) (1 point) What is the inverse of the transformation matrix in part (b) of this question?

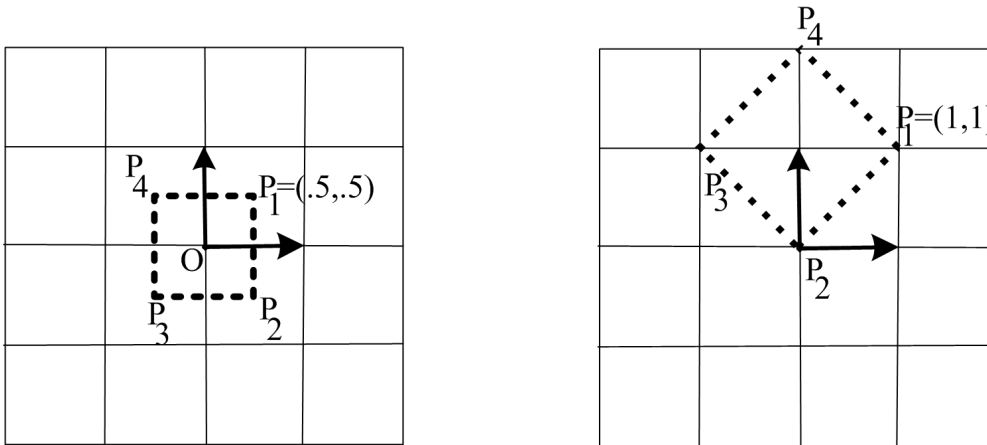
This is  $S^{-1}T^{-1}$  or

$$S^{-1}T^{-1} = \begin{pmatrix} 1 & 0 & 0 & -2 \\ 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(d) (1 point) Give the sequence of OpenGL transformations that would produce the same transformation matrix as in part (a) of this question.

`glRotatef( 90, 0, 0, 1 ); glScalef( 0.5, 0.5, 0.5 );`

4. (3 points) Write down the 2D transformation matrix that maps the unit square centered at the origin as shown on the left to the square on the right in the figure below. Show your work.



There are two ways to do this. The first way is to translate, then scale, then rotate by -45 degrees (when operating the global coordinate frame). So the transformation matrix is

$$R * S * T$$

where

$$T = \begin{pmatrix} 1 & 0 & -0.5 \\ 0 & 1 & 0.5 \\ 0 & 0 & 1 \end{pmatrix}$$

$$S = \begin{pmatrix} \sqrt{2} & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R = \begin{pmatrix} \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ -\sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The second is to scale, rotate by -45 degrees, and translate (when operating in the global coordinate frame).

$$T * R * S$$

where

$$S = \begin{pmatrix} \sqrt{2} & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R = \begin{pmatrix} \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ -\sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

5. (4 points) Answer yes/no and provide a short explanation. All the transformations are in 3D.

(a) (1 point) Does perspective transformation preserve parallel lines?

No. Unless the lines are parallel to the horizon they will meet at the horizon after the transformation.

(b) (1 point) Is shear \* translate = translate \* shear?

No. Consider the shear  $S$  and the translate  $T$

$$S = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Then  $S * T \neq T * S$

(c) (1 point) Is shear1 \* shear2 = shear2 \* shear1?

No. Consider the shear  $S1$  and the shear  $S2$

$$S1 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$S2 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Then  $S1 * S2 \neq S2 * S1$

(d) (1 point) Does shear preserve lengths?

No. Consider a square and the shear  $S_2$  from the previous part. Applying the shear  $S_2$  to the square changes the length of the diagonal of the square.

6. (3 points) Given the triangle  $T$  with vertices  $P_1 = (0, 0, 0), P_2 = (2, 0, 0), P_3 = (1, 1, 0)$  and the transformation

$$S = \begin{pmatrix} 1 & -0.5 & 0 & 0 \\ 0.5 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(a) (2 points) Compute the vertices of the triangle after applying the transformation  $S$  to it.

Multiply each point by  $S$ .

$$S * \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$S * \begin{pmatrix} 2 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

$$S * \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 2.5 \\ 0 \\ 1 \end{pmatrix}$$

So the  $P_1 = (0, 0, 0), P_2 = (2, 1, 0), P_3 = (0.5, 2.5, 0)$ .

(b) (1 point) Compute the normal of the triangle before and after applying the transformation  $S$  to it.

Before the transformation

$$(P_2 - P_1) \times (P_3 - P_1) = (0, 0, 2)$$

After the transformation

$$(P_2 - P_1) \times (P_3 - P_1) = (0, 0, 4.5)$$

7. (4 points) BONUS: Given a non-convex planar polygon  $P = P_1, P_2, \dots, P_n$  and a point  $P$  in 2D describe an algorithm for testing if the point is inside/outside the polygon. Try to make your algorithm as efficient as possible.

Construct a ray  $R$  starting at point  $P$ . Compute a ray-edge intersection test with all the edges of the polygon and  $R$ . If the number of intersections is odd, then  $P$  is in the polygon. If it is even, then  $P$  is outside of the polygon. Simulate the algorithm on paper to understand why this works. Note that there is a possibility of intersecting a single point which can yield a false negative because the point belongs to two segments. We can get around this by offsetting the ray in a nonparallel direction by some very small constant  $\epsilon$ . If both  $R$  and  $R + \epsilon$  give the same result, then we can assume the result is correct. If they don't have the same result, then assume the ray with the lesser number of intersections is correct.

Note that this is much more efficient than triangulating a nonconvex polygon and testing to see if the point is in any of the triangles.