

# Lab Report

Ritobroto Maitra (1301CS50)

Week 3

19/01/2016

---

## ■ Title

- Applying transformations to basic primitives (a line, a circle and a polygon).

---

## Procedure

We use a combination of in built matrix transformations along with some hard coded matrix multiplications to do the requisite modeling transformations.

---

## Transformation Matrices

1. Scaling

$$\begin{bmatrix} 0.33 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2. Reflection

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3. Shear

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

4. Rotation

$$\begin{bmatrix} \cos(45) & -\sin(45) & 0 \\ \sin(45) & \cos(45) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

5. Translation

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0.1 \\ 0 & 0 & 1 \end{bmatrix}$$

---

## Drawing the Basic Shapes in OpenGL

The following are the functional parts of the code required to plot the shapes.

### Line

We create two points and draw a line between them.

```
1  glVertex3f(0.0, 0.0, 0.0);
2  glVertex3f(0.5, -0.5, 0);
```

## Circle

We create a 0.5 radius circle using  $\sin(i)$  and  $\cos(i)$  values as follows.

```
1  for(i=0;i<10*3.14;i+=0.0001){
2      glVertex3f(cos(i)/2, sin(i)/2, 0.0);
3  }
```

## Polygon

The polygon is created using the following lines.

```
1  glBegin(GL_LINES);
2  glVertex3f(0.5, 0.5, 0.0);
3  glVertex3f(0.0, 0.0, 0.0);
4  glVertex3f(0.0, 0.0, 0.0);
5  glVertex3f(-0.5, 0.5, 0.0);
6  glVertex3f(-0.5, 0.5, 0.0);
7  glVertex3f(-0.7, -0.5, 0.0);
8  glVertex3f(-0.7, -0.5, 0.0);
9  glVertex3f(-0.2, -0.6, 0.0);
10 glVertex3f(-0.2, -0.6, 0.0);
11 glVertex3f(0.5, 0.5, 0.0);
```

## Transformation Code in OpenGL

The transformation is fed by an input of options which is dealt by the following switch case mechanism.

```
1  static void Key(unsigned char key, int x, int y)
2  {
3      GLfloat m[] = {1, 0, 0, 0, 3, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1};
4      GLfloat n[] = {1, 0, 0, 0, -3, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1};
5      glMatrixMode(GL_MODELVIEW);
6      switch (key) {
7          case '1':
8              glScalef(0.33, 1, 1);
9              glutPostRedisplay();
10             break;
11             case '2':
12                 glScalef(3, 1, 1);
13                 glutPostRedisplay();
14                 break;
15             case '3':
16                 glRotatef(45, 0, 0, 1);
17                 glutPostRedisplay();
18                 break;
19             case '4':
20                 glMultMatrixf(m);
21                 glutPostRedisplay();
22                 break;
23             case '5':
24                 glMultMatrixf(n);
25                 glutPostRedisplay();
26                 break;
27             case '6':
28                 glScalef(-1, 1, 1);
```

```

29     glutPostRedisplay();
30     break;
31 case '7':
32     glTranslatef(0, 0.1, 0);
33     glutPostRedisplay();
34     break;
35 case '8':
36     glTranslatef(0, -0.1, 0);
37     glutPostRedisplay();
38     break;
39     case 27:
40     exit(0);
41 }
42 }

```

## Transformation Code in MATLAB

```

1 while in ~= 0
2 in = input(prompt);
3 if in==1
4     % scaling
5     T = [2, 0, 0; 0, 1.5, 0; 0, 0, 1];
6     pts = T*pts;
7     plot(pts(1,:), pts(2,:), 'r*-');
8     axis([-10, 10, -10, 10]);
9     display(pts)
10 elseif in==-1
11     % rev scaling
12     T = [0.5, 0, 0; 0, 2/3, 0; 0, 0, 1];
13     pts = T*pts;
14     plot(pts(1,:), pts(2,:), 'r*-');
15     axis([-10, 10, -10, 10]);
16     display(pts)
17 elseif in==2
18     % reflection
19     T = [-1, 0, 0; 0, 1, 0; 0, 0, 1];
20     pts = T*pts;
21     plot(pts(1,:), pts(2,:), 'r*-');
22     axis([-10, 10, -10, 10]);
23     display(pts)
24 elseif in==3
25     % shear
26     T = [1, 1, 0; 0, 1, 0; 0, 0, 1];
27     pts = T*pts;
28     plot(pts(1,:), pts(2,:), 'r*-');
29     axis([-10, 10, -10, 10]);
30     display(pts)
31 elseif in==3
32     % rev shear
33     T = [1, -1, 0; 0, 1, 0; 0, 0, 1];
34     pts = T*pts;
35     plot(pts(1,:), pts(2,:), 'r*-');
36     axis([-10, 10, -10, 10]);
37     display(pts)
38 elseif in==4
39     % rotation
40     T = [cosd(45), -sind(45), 0; sind(45), cosd(45), 0; 0, 0, 1];
41     pts = T*pts;
42     plot(pts(1,:), pts(2,:), 'r*-');
43     axis([-10, 10, -10, 10]);
44     display(pts)
45 elseif in==5
46     % translation
47     T = [1, 0, 1; 0, 1, 0.5; 0, 0, 1];
48     pts = T*pts;
49     plot(pts(1,:), pts(2,:), 'r*-');
50     axis([-10, 10, -10, 10]);
51     display(pts)
52 elseif in==5
53     % rev translation
54     T = [1, 0, -1; 0, 1, -0.5; 0, 0, 1];
55     pts = T*pts;
56     plot(pts(1,:), pts(2,:), 'r*-');

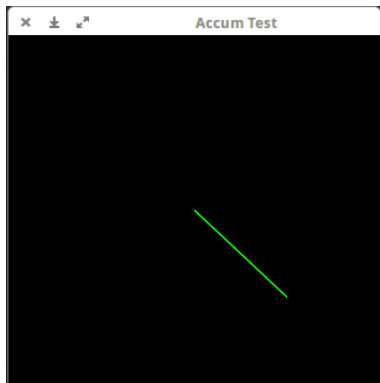
```

```

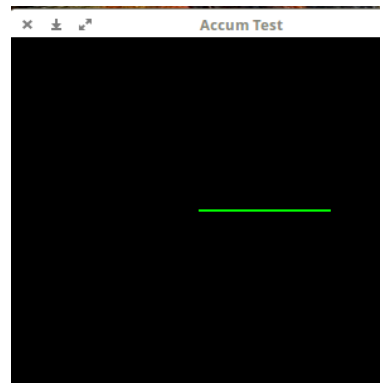
57     axis([-10, 10, -10, 10]);
58     display(pts)
59 end

```

## Examples

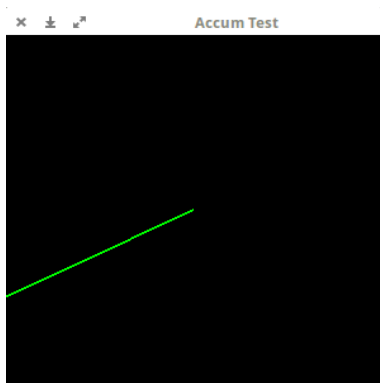


(a) Initial Line

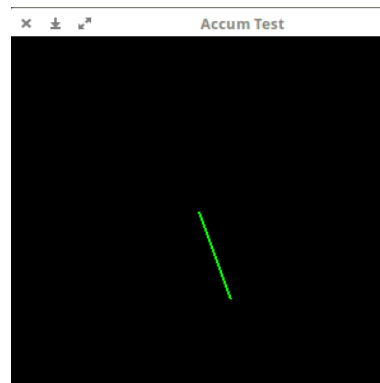


(b) Line : Rotation

FIGURE 1 – Transformations on Line : Rotation

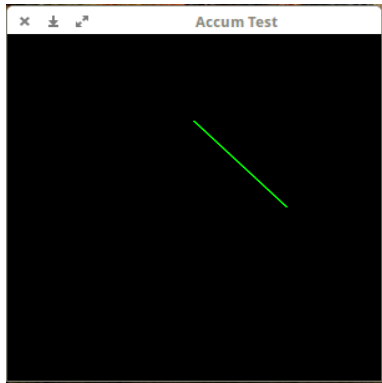


(a) Line : Shear

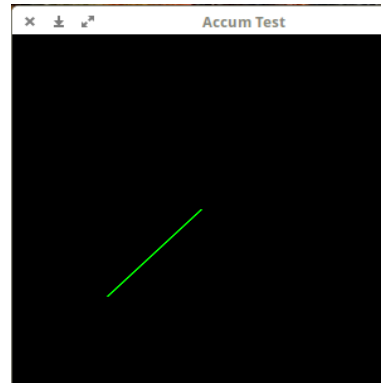


(b) Line Scale

FIGURE 2 – Transformations on Line : Shear, Scale

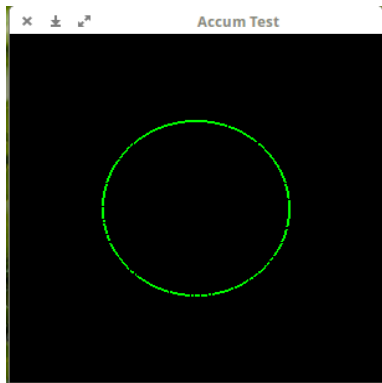


(a) Line : Translate

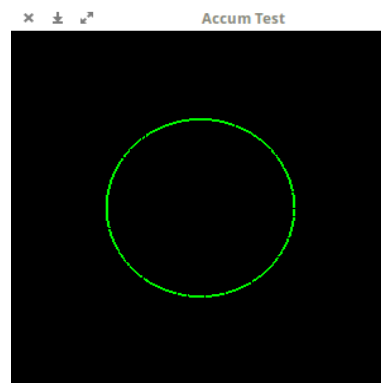


(b) Line : Reflect

FIGURE 3 – Transformations on Line : Translate, Reflect

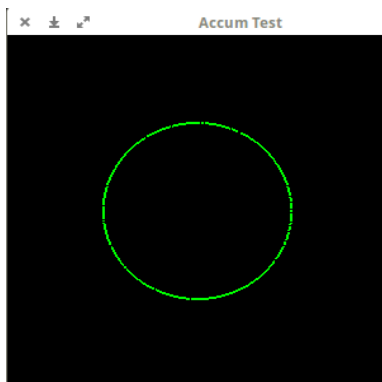


(a) Initial Circle

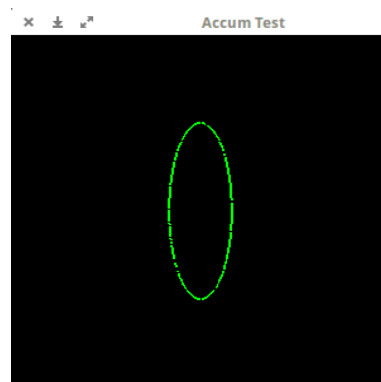


(b) Circle : Rotation

FIGURE 4 – Transformations on Circle : Rotation

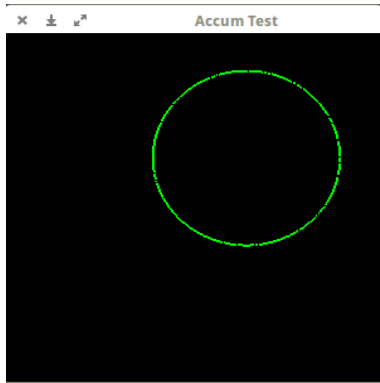


(a) Circle : Shear

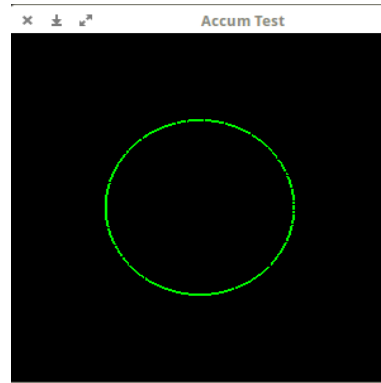


(b) Line Scale

FIGURE 5 – Transformations on Circle : Shear, Scale

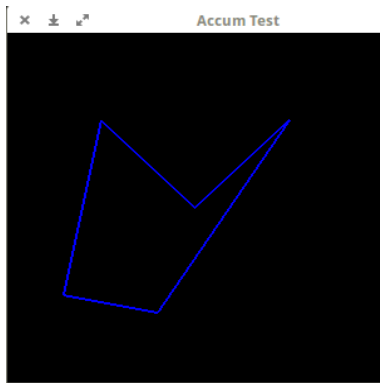


(a) Circle : Translate

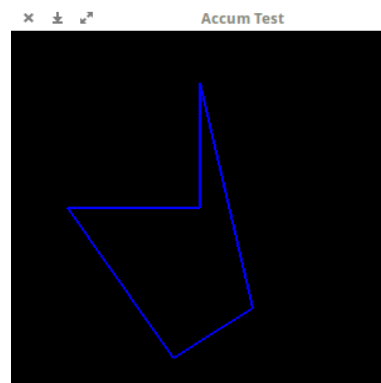


(b) Circle : Reflect

FIGURE 6 – Transformations on Circle : Translate, Reflect

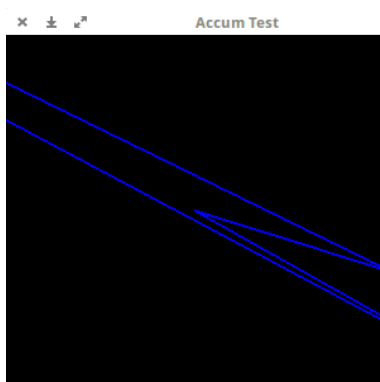


(a) Initial polygon

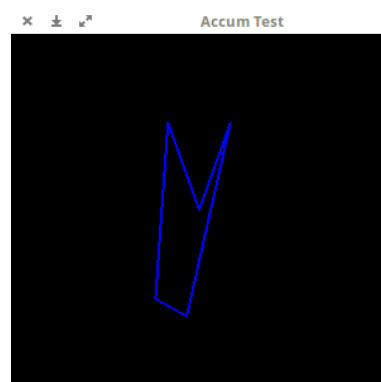


(b) polygon : Rotation

FIGURE 7 – Transformations on polygon : Rotation

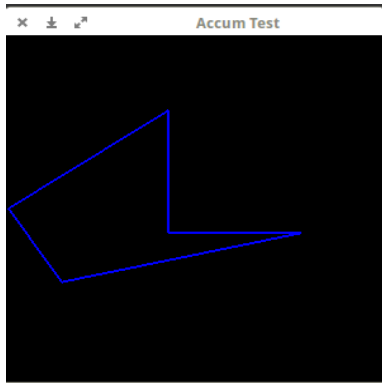


(a) Polygon : Shear

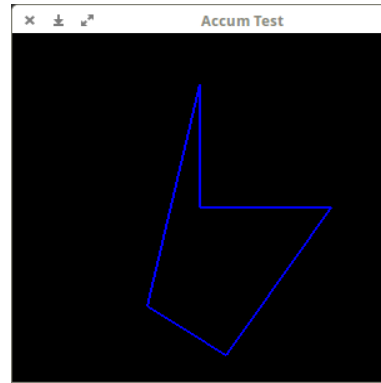


(b) Polygon : Scale

FIGURE 8 – Transformations on Polygon : Shear, Scale

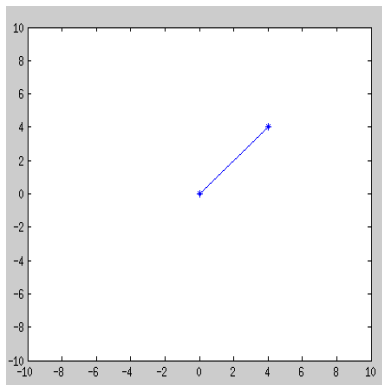


(a) Polygon : Translate

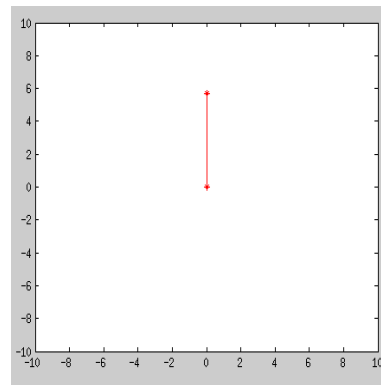


(b) Polygon : Reflect

FIGURE 9 – Transformations on Polygon : Translate, Reflect

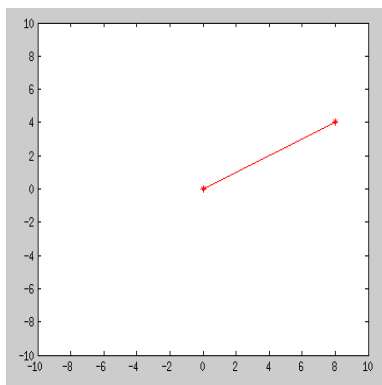


(a) Initial Line

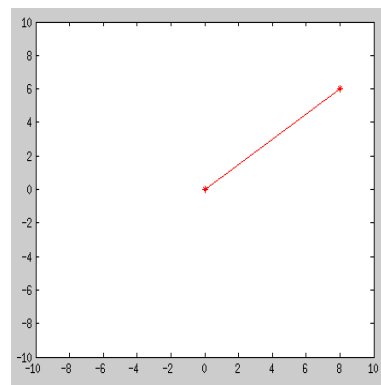


(b) Line : Rotation

FIGURE 10 – Transformations on Line : Rotation

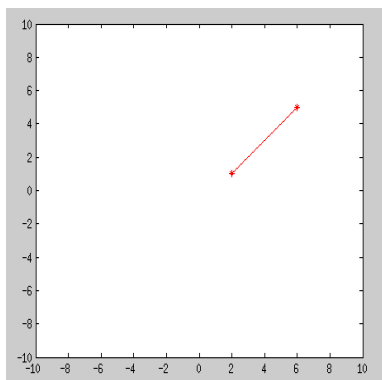


(a) Line : Shear

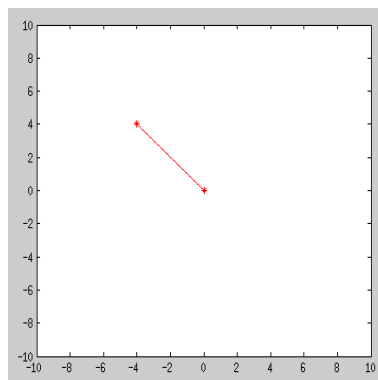


(b) Line Scale

FIGURE 11 – Transformations on Line : Shear, Scale

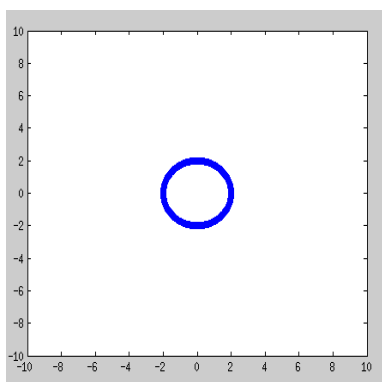


(a) Line : Translate

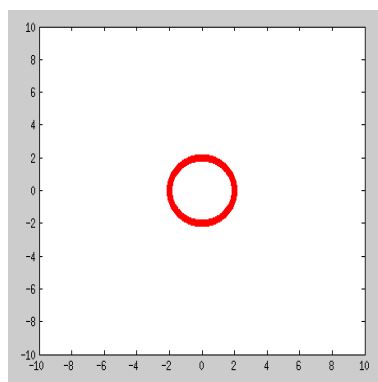


(b) Line : Reflect

FIGURE 12 – Transformations on Line : Translate, Reflect

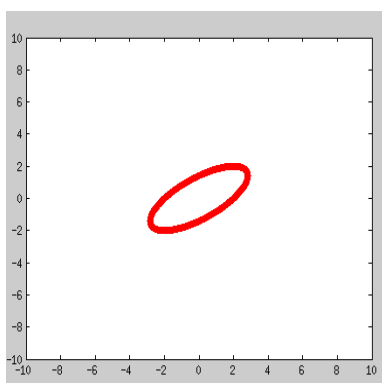


(a) Initial Circle

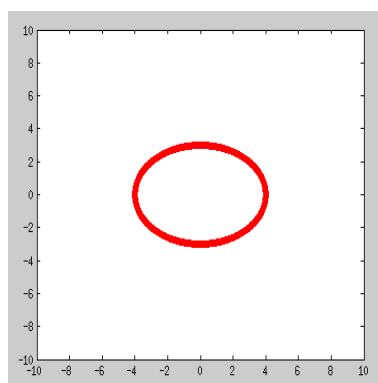


(b) Circle : Rotation

FIGURE 13 – Transformations on Circle : Rotation



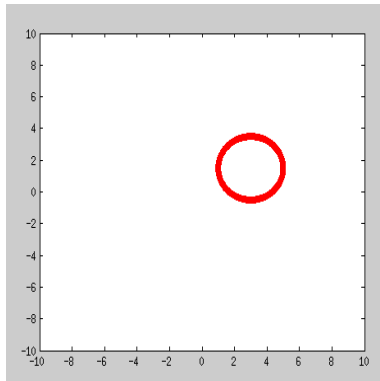
(a) Circle : Shear



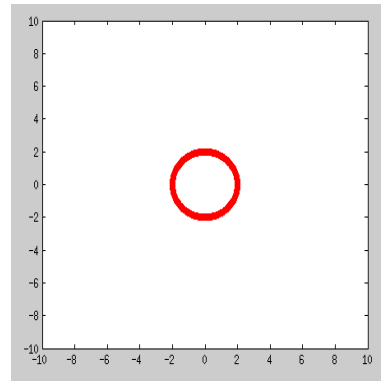
(b) Circle : Scale

FIGURE 14 – Transformations on Circle : Shear, Scale



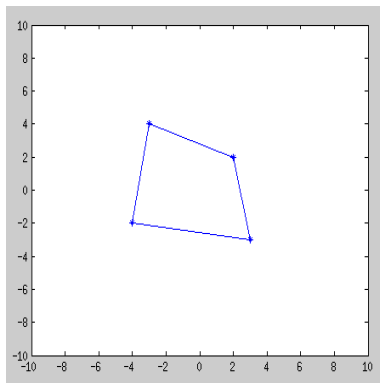


(a) Circle : Translate

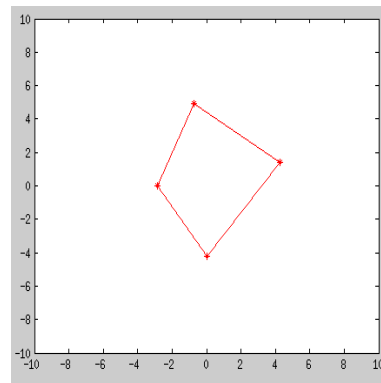


(b) Circle : Reflect

FIGURE 15 – Transformations on Circle : Translate, Reflect

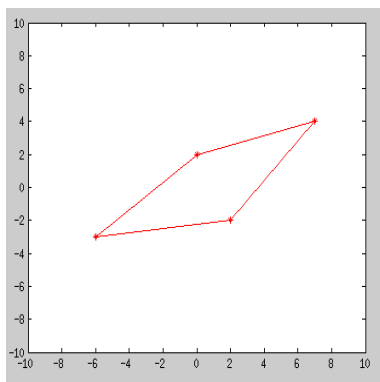


(a) Initial polygon

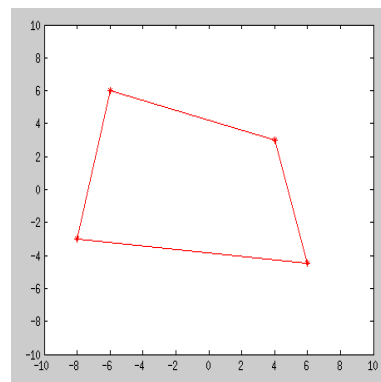


(b) Polygon : Rotation

FIGURE 16 – Transformations on Polygon : Rotation

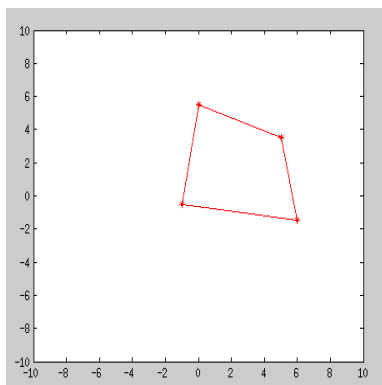


(a) Polygon : Shear

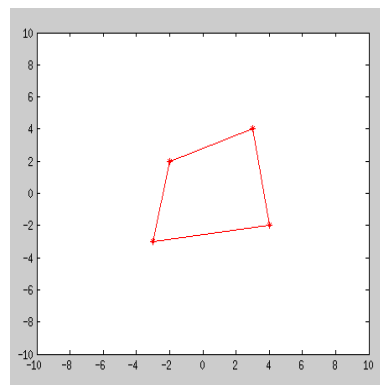


(b) Polygon : Scale

FIGURE 17 – Transformations on Polygon : Shear, Scale



(a) Polygon : Translate



(b) Polygon : Reflect

FIGURE 18 – Transformations on Polygon : Translate, Reflect