Computer Graphics Lab 5: Two Dimensional Transformations

Bibek Subedi

January 27, 2026

Contents

Aim

To apply basic 2D transformations such as Translation, Scaling, Rotation, Reflection, and Shearing for a given 2D object.

Description

In this experiment, we perform 2D transformations on a 2D object, such as a line segment.
The 2D transformations include:

1. **Translation:** Moving the object from one position to another along a straight line.
2. **Scaling:** Changing the size of an object by scaling factors along the X and Y axes.
3. **Rotation:** Rotating the object about the origin by an angle θ.
4. **Reflection:** Producing a mirror image of the object about a given axis.
5. **Shearing:** Distorting the shape of the object along X or Y axis.

Algorithms

1. Translation

$$x' = x + t_x, \quad y' = y + t_y$$

2. Scaling

$$x' = x \cdot s_x, \quad y' = y \cdot s_y$$

3. Rotation

$$x' = x \cos\theta - y \sin\theta, \quad y' = x \sin\theta + y \cos\theta$$

4. Reflection

$$\text{X-axis: } (x, -y), \quad \text{Y-axis: } (-x, y), \quad \text{Origin: } (-x, -y)$$

5. Shearing

$$x' = x + sh_x \cdot y, \quad y' = y + sh_y \cdot x$$

C Programs

Translation Program (lab5_1.cpp)

```c
#include <graphics.h>
#include <stdio.h>
#include <conio.h>

int main() {
    int gd = DETECT, gm;
    int x1, y1, x2, y2, tx, ty;

    printf("Enter x1, y1, x2, y2: ");
    scanf("%d %d %d %d", &x1, &y1, &x2, &y2);
    printf("Enter tx, ty: ");
    scanf("%d %d", &tx, &ty);

    initgraph(&gd, &gm, "");

    setcolor(WHITE);
    line(x1, y1, x2, y2);
    outtextxy(x1, y1 - 10, "Original");

    setcolor(YELLOW);
    line(x1 + tx, y1 + ty, x2 + tx, y2 + ty);
    outtextxy(x1 + tx, y1 + ty - 10, "Translated");

    getch();
    closegraph();
    return 0;
}
```

Scaling Program (lab5_2.cpp)

```c
#include <graphics.h>
#include <stdio.h>
#include <conio.h>

int main() {
    int gd = DETECT, gm;
    int x1, y1, x2, y2;
    float sx, sy;

    printf("Enter x1, y1, x2, y2: ");
    scanf("%d␣%d␣%d␣%d", &x1, &y1, &x2, &y2);
    printf("Enter␣sx,␣sy:␣");
    scanf("%f␣%f", &sx, &sy);

    initgraph(&gd, &gm, "");

    setcolor(WHITE);
    line(x1, y1, x2, y2);
    outtextxy(x1, y1 - 10, "Original");

    setcolor(GREEN);
    line(x1*sx, y1*sy, x2*sx, y2*sy);
    outtextxy(x1*sx, y1*sy - 10, "Scaled");

    getch();
    closegraph();
    return 0;
}
```

Rotation Program (lab5_3.cpp)

```
#include <graphics.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>

int main() {
    int gd = DETECT, gm;
    int x1, y1, x2, y2;
    float theta;
    float rad;

    printf("Enter␣x1,␣y1,␣x2,␣y2:␣");
    scanf("%d␣%d␣%d␣%d", &x1, &y1, &x2, &y2);
    printf("Enter␣rotation␣angle␣(degrees):␣");
    scanf("%f", &theta);

    rad = theta * 3.14159 / 180;

    initgraph(&gd, &gm, "");

    setcolor(WHITE);
    line(x1, y1, x2, y2);
    outtextxy(x1, y1 - 10, "Original");
    setcolor(BLUE);
    line((int)(x1*cos(rad)-y1*sin(rad)), (int)(x1*sin(rad)+y1*cos(rad)),
         (int)(x2*cos(rad)-y2*sin(rad)), (int)(x2*sin(rad)+y2*cos(rad)));
    outtextxy((int)(x1*cos(rad)-y1*sin(rad)), (int)(x1*sin(rad)+y1*cos(rad)) -

    getch();
    closegraph();
    return 0;
}
```

Reflection Program (lab5_4.cpp)

```
#include <graphics.h>
#include <stdio.h>
#include <conio.h>

int main() {
    int gd = DETECT, gm;
    int x1, y1, x2, y2;

    printf("Enter x1, y1, x2, y2: ");
    scanf("%d %d %d %d", &x1, &y1, &x2, &y2);

    initgraph(&gd, &gm, "");

    setcolor(WHITE);
    line(x1, y1, x2, y2);
    outtextxy(x1, y1 - 10, "Original");

    setcolor(MAGENTA);
    line(x1, -y1, x2, -y2); // reflection about X-axis
    outtextxy(x1, -y1 - 10, "Reflected");

    getch();
    closegraph();
    return 0;
}
```

Shearing Program (lab5_5.cpp)

```
#include <graphics.h>
#include <stdio.h>
```

```c
#include <conio.h>

int main() {
    int gd = DETECT, gm;
    int x1, y1, x2, y2;
    float shx, shy;

    printf("Enter x1, y1, x2, y2: ");
    scanf("%d %d %d %d", &x1, &y1, &x2, &y2);
    printf("Enter shx, shy: ");
    scanf("%f %f", &shx, &shy);

    initgraph(&gd, &gm, "");

    setcolor(WHITE);
    line(x1, y1, x2, y2);
    outtextxy(x1, y1 - 10, "Original");

    setcolor(CYAN);
    line(x1 + shx*y1, y1 + shy*x1, x2 + shx*y2, y2 + shy*x2);
    outtextxy(x1 + shx*y1, y1 + shy*x1 - 10, "Sheared");

    getch();
    closegraph();
    return 0;
}
```

Results

All programs successfully displayed the original object and its transformed version on the screen.

Conclusion

We successfully implemented basic 2D transformations and observed the effects of Translation, Scaling, Rotation, Reflection, and Shearing on 2D objects.