International Islamic University Chittagong

Department of Computer Science and Engineering

Lab Report On : Implement Southerland Hodgeman Polygon Clipping Algorithm.

Course Title: Computer Graphics Lab

Course Code: CSE-4742

Submitted to:

Sultana Tasnim Jahan

Lecturer

Dept of CSE(IIUC)

Submitted By:

Name	Jannatul Nayeem
ID	C223293
Semester	7th
Section	7CF

Date of submission:20-11-2025

1. Problem Name

Implementation of Sutherland–Hodgman Polygon Clipping Algorithm

Description

In computer graphics, polygon clipping is the process of cutting a polygon so that only the part inside a specified clipping window (usually a rectangle) remains.
The **Sutherland–Hodgman algorithm** is a widely used polygon-clipping algorithm that clips a polygon against each boundary of the clipping rectangle one at a time.

It works by taking a list of polygon vertices and successively clipping them against:

1. Left boundary
2. Right boundary
3. Bottom boundary
4. Top boundary

At each stage, a new vertex list is created until the final clipped polygon is produced.

It is especially useful for convex clipping windows.

Algorithm (steps)

Given a polygon with vertices in order:
V = [v_1, v_2, v_3, …, v_n] and a rectangular clipping window:
(xmin, ymin) to (xmax, ymax)

We clip polygon edges against each boundary one by one.

For each edge of polygon (start = S, end = P):

1. **Case 1: S is inside & P is inside → output P**
2. **Case 2: S is inside & P is outside → output intersection**
3. **Case 3: S is outside & P is inside → output intersection + P**
4. **Case 4: S is outside & P is outside → output nothing**

Repeat this for each boundary until all four boundaries are processed.

The final list is the clipped polygon.

Code:

```
import matplotlib.pyplot as plt
def inside(p, boundary, value):
    x, y = p
    if boundary == "LEFT":
        return x >= value
    elif boundary == "RIGHT":
        return x <= value
    elif boundary == "BOTTOM":
        return y >= value
    elif boundary == "TOP":
        return y <= value
def intersect(s, p, boundary, value):
    x1, y1 = s
    x2, y2 = p
    if x1 != x2:
        m = (y2 - y1) / (x2 - x1)
    else:
        m = None
    if boundary == "LEFT":
        x = value
        y = y1 + (value - x1) * m if m is not None else y1
        return (x, y)
    elif boundary == "RIGHT":
        x = value
        y = y1 + (value - x1) * m if m is not None else y1
        return (x, y)
    elif boundary == "BOTTOM":
        y = value
        if m is None:
            x = x1
        else:
            x = x1 + (value - y1) / m
        return (x, y)
    elif boundary == "TOP":
        y = value
        if m is None:
            x = x1
        else:
            x = x1 + (value - y1) / m
        return (x, y)
def clip_polygon(polygon, xmin, ymin, xmax, ymax):
    boundaries = [
        ("LEFT", xmin),
```

```
        ("RIGHT", xmax),
        ("BOTTOM", ymin),
        ("TOP", ymax)
    ]
    output = polygon
    for boundary, value in boundaries:
        new_output = []
        for i in range(len(output)):
            s = output[i - 1]
            p = output[i]
            if inside(p, boundary, value):
                if inside(s, boundary, value):
                    new_output.append(p)
                else:
                    new_output.append(intersect(s, p, boundary, value))
                    new_output.append(p)
            else:
                if inside(s, boundary, value):
                    new_output.append(intersect(s, p, boundary, value))
        output = new_output
    return output
polygon = [(1,2), (4,7), (8,6), (7,3), (4,1)]
xmin, ymin = 2, 2
xmax, ymax = 7, 6
clipped = clip_polygon(polygon, xmin, ymin, xmax, ymax)
print("Original Polygon:", polygon)
print("Clipped Polygon:", clipped)
plt.figure(figsize=(8,6))
xp, yp = zip(*(polygon + [polygon[0]]))
plt.plot(xp, yp, 'r--', label="Original Polygon")
if clipped:
    xc, yc = zip(*(clipped + [clipped[0]]))
    plt.plot(xc, yc, 'b-', linewidth=2, label="Clipped Polygon")
plt.plot([xmin, xmax, xmax, xmin, xmin],
         [ymin, ymin, ymax, ymax, ymin],
         'k-', linewidth=2, label="Clipping Window")

plt.title("Sutherland-Hodgman Polygon Clipping")
plt.legend()
plt.grid(True)
plt.show()
```

```python
import matplotlib.pyplot as plt

def inside(p, boundary, value):
    x, y = p
    if boundary == "LEFT":
        return x >= value
    elif boundary == "RIGHT":
        return x <= value
    elif boundary == "BOTTOM":
        return y >= value
    elif boundary == "TOP":
        return y <= value

def intersect(s, p, boundary, value):
    x1, y1 = s
    x2, y2 = p

    if x1 != x2:
        m = (y2 - y1) / (x2 - x1)
    else:
        m = None

    if boundary == "LEFT":
        x = value
        y = y1 + (value - x1) * m if m is not None else y1
        return (x, y)

    elif boundary == "RIGHT":
        x = value
        y = y1 + (value - x1) * m if m is not None else y1
        return (x, y)

    elif boundary == "BOTTOM":
```

Output:

```python
plt.legend()
plt.grid(True)
plt.show()
```

```
Original Polygon: [(1, 2), (4, 7), (8, 6), (7, 3), (4, 1)]
Clipped Polygon: [(2, 2), (2, 3.666666666666667), (3.4, 6), (7, 6), (7, 3.0), (7, 3), (5.5, 2)]
```