Algoritmo de Recorte

DCC703 - Computação Gráfica (2024.2)
Prof. - Luciano Ferreira Silva
Aluno - Paulo Ferreira da Silva Júnior - 2019034400

Introdução

O recorte de polígonos é uma operação fundamental em computação gráfica, sendo utilizado para exibir apenas as partes visíveis de uma cena dentro de uma região específica. O **algoritmo de Sutherland-Hodgman** é uma das técnicas mais conhecidas para essa tarefa, permitindo recortar polígonos arbitrários contra uma janela retangular.

Neste relatório, implementamos e analisamos a execução do algoritmo de recorte, observando sua eficácia em diferentes casos de polígonos.

---

1. Algoritmo de Recorte de Sutherland-Hodgman

Descrição do Algoritmo

O algoritmo de **Sutherland-Hodgman** funciona percorrendo todas as arestas do polígono original e comparando-as com as arestas da janela de recorte. O processo é aplicado iterativamente para cada borda da região de recorte, conforme as seguintes regras:

1. **Se um ponto está dentro da janela**, ele é mantido no polígono recortado.
2. **Se um segmento cruza a borda da janela**, o ponto de interseção entre a aresta do polígono e a borda da janela é calculado e adicionado.
3. **Se um ponto está fora da janela**, ele é removido do polígono resultante.

Esse processo é repetido para as **quatro bordas** da janela de recorte (esquerda, direita, superior e inferior), garantindo que ao final o polígono resultante esteja completamente dentro da região visível.

Fórmula para Interseção

Para calcular a interseção entre um segmento de reta e a borda da região de recorte, utilizamos a fórmula:

(xint,yint) = (x1+ (x2−x1) (yclip−y1) /y2−y1, yclip)

onde:

- (x1,y1) e (x2,y2) são os pontos da aresta do polígono.
- yclip é o valor de recorte da janela (superior ou inferior).
- Se a interseção for com uma borda vertical, os papéis de x e y são invertidos.

---

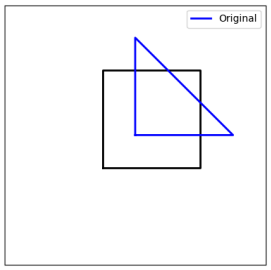
2. Implementação

O código a seguir implementa o **algoritmo de Sutherland-Hodgman** em Python utilizando a biblioteca **Matplotlib** para visualização e **NumPy** para manipulação de pontos.

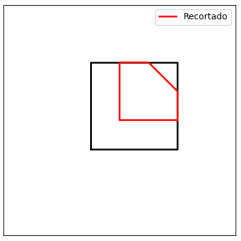
```
def inside(point, clip_edge):
    x, y = point
    edge_x1, edge_y1, edge_x2, edge_y2 = clip_edge
    return x >= edge_x1 if edge_x1 == xmin else x <= edge_x1

def intersection(p1, p2, clip_edge):
    x1, y1 = p1
    x2, y2 = p2
    edge_x1, edge_y1, edge_x2, edge_y2 = clip_edge

    if edge_x1 == edge_x2:  # Aresta vertical
        x_int = edge_x1
        m = (y2 - y1) / (x2 - x1) if x2 != x1 else 0
        y_int = y1 + m * (x_int - x1)
    else:  # Aresta horizontal
        y_int = edge_y1
        m = (y2 - y1) / (x2 - x1) if x2 != x1 else 0
        x_int = x1 + (y_int - y1) / m if m != 0 else x1

    return (x_int, y_int)

def clip_polygon(polygon, clip_rect):
    x_min, x_max, y_min, y_max = clip_rect
    clip_edges = [
```

```
    (x_min, y_min, x_min, y_max),  # Esquerda
    (x_min, y_max, x_max, y_max),  # Superior
    (x_max, y_max, x_max, y_min),  # Direita
    (x_max, y_min, x_min, y_min)   # Inferior
]

clipped_polygon = polygon[:]
for clip_edge in clip_edges:
    new_polygon = []
    prev_point = clipped_polygon[-1]
    for curr_point in clipped_polygon:
        if inside(curr_point, clip_edge):
            if not inside(prev_point, clip_edge):
                intersec = intersection(prev_point, curr_point, clip_edge)
                new_polygon.append(intersec)
            new_polygon.append(curr_point)
        elif inside(prev_point, clip_edge):
            intersec = intersection(prev_point, curr_point, clip_edge)
            new_polygon.append(intersec)
        prev_point = curr_point
    clipped_polygon = new_polygon

return clipped_polygon
```

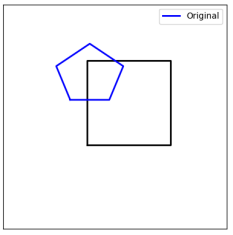
3. Resultados Obtidos



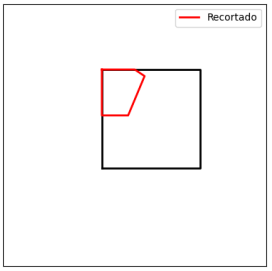
Poligono A inteiro



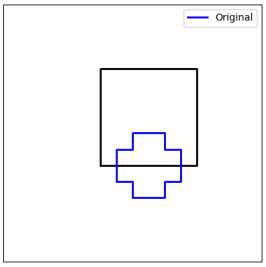
Poligono A cortado



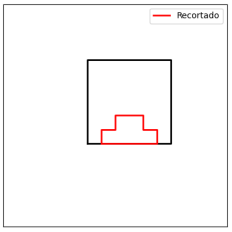
Poligono B inteiro



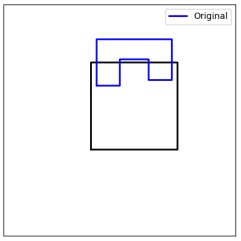
Poligono B cortado



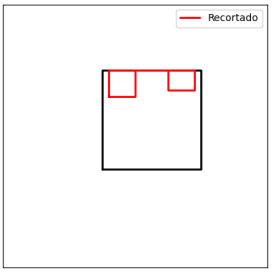
Poligono C inteiro



Poligono C cortado



Poligono D inteiro



Poligono D cortado

---

4. Comparação e Conclusão

Critério	Sutherland-Hodgman
Eficiência	Alta
Precisão	Alta
Aplicação	Ideal para recorte contra regiões retangulares

O algoritmo de **Sutherland-Hodgman** é eficiente e amplamente utilizado em aplicações gráficas para recortar polígonos contra regiões retangulares. No entanto, ele não funciona para janelas de recorte arbitrárias, onde algoritmos como **Weiler-Atherton** são mais adequados.
