



Centro de Enseñanza Técnica Industrial

Desarrollo de Software

Recorte de regiones convexas

Jesús Alberto Aréchiga Carrillo

22310439

6N

Profesor

José Francisco Pérez Reyes

Mayo 2025

Guadalajara, Jalisco

Introducción

El recorte es el proceso de determinar qué región del espacio se encuentra dentro de otra región del espacio. Esto es especialmente importante en áreas de estudio como gráficos por computadora y simulación física. Por ejemplo, al representar un mundo en la pantalla, es mejor saber qué aparecerá en ella antes de procesar cualquier dato. Esto permite ignorar muchos datos ajenos, procesando y presentando únicamente lo que verá el usuario.

Existen varios algoritmos para el recorte de regiones convexas; uno de los más conocidos es el de Sutherland–Hodgman. Su funcionamiento es el siguiente: a partir del conjunto inicial de vértices del polígono, se recorta contra la primera frontera para obtener una nueva secuencia de vértices. A continuación, esta secuencia se recorta contra la segunda frontera, y así sucesivamente hasta procesar todas las fronteras. Este método resulta especialmente útil para recortar bucles de línea contra planos convexas. Dado un polígono definido por una lista de vértices y un conjunto de planos que delimitan la región de recorte, el algoritmo produce como salida una nueva lista de vértices que describe el polígono resultante, el cual queda completamente contenido dentro de la región convexa.

Desarrollo

El algoritmo opera iterativamente sobre las aristas de la región convexa de recorte, aplicando en cada paso un “filtro” que construye un nuevo polígono de salida a partir del polígono de entrada y una frontera concreta. Supongamos que la región de recorte es un polígono convexo de k aristas y nuestro polígono de entrada tiene n vértices:

1. Inicialización

Se parte de la lista original de vértices entrada = $[v_0, v_1, \dots, v_{n-1}]$.

2. Recorte contra cada arista

Para cada arista E_i de la ventana de recorte:

1. Se crea una lista vacía salida.
2. Se toma como punto anterior S = último vértice de entrada.
3. Para cada vértice actual P en entrada:
 - Si P está **dentro** de E_i :
 - Si S estaba **fuera**, calcular el punto de intersección I entre el segmento $S \rightarrow P$ y la recta de E_i , y añadir I a salida.
 - Añadir también P a salida.

- Si P está **fuera** y S estaba **dentro**, calcular e insertar primero la intersección I en salida, pero no P.
- Actualizar $S \leftarrow P$.

4. Reemplazar entrada \leftarrow salida y continuar con la siguiente arista.

Al terminar las k iteraciones, entrada contiene el polígono resultante completamente dentro de la región convexa de recorte.

Implementación en Java

```
public List<Point> sutherlandHodgman(List<Point> poly, List<Edge> clipEdges) {
    List<Point> output = new ArrayList<>(poly);
    for (Edge edge : clipEdges) {
        List<Point> input = output;
        output = new ArrayList<>();
        Point S = input.get(input.size() - 1);
        for (Point P : input) {
            boolean inP = edge.isInside(P), inS = edge.isInside(S);
            if (inP) {
                if (!inS) output.add(edge.intersect(S, P));
                output.add(P);
            } else if (inS) {
                output.add(edge.intersect(S, P));
            }
            S = P;
        }
    }
    return output;
}
```

Edge es una interfaz que implementa `isInside(Point)` y `intersect(Point,Point)` para cada lado de la ventana de recorte.

Conclusión

El método de Sutherland–Hodgman ofrece una solución limpia, robusta y de implementación directa para el recorte de polígonos contra regiones convexas. Su sencillez conceptual y su coste lineal moderado lo convierten en la opción de referencia en motores gráficos 2D y etapas iniciales de renderizado de escenas 3D.