

Álvaro Manchado Berzosa
Notas sobre el salvapantallas de Cierres Convexos

Cierre Convexo:

La aplicación es un salvapantallas completo, con todas las características propias de este tipo de archivos.

- Haciendo doble click en el archivo se ejecutará el salvapantallas.
- Haciendo clic con el botón derecho sobre el, aparecerán algunas opciones especiales:
 - Probar: Ejecuta la aplicación.
 - Configurar: Muestra el dialogo de configuración de parámetros de la aplicación.
 - Instalar: Instala el salvapantallas como activo de Windows y muestra el dialogo de Propiedades de pantalla. Desde el cual se puede acceder a la configuración y vista previa.

Además, se incluye un modo especial de funcionamiento que permite interactuar con los algoritmos, suministrar puntos, etc. Para acceder a este modo interactivo pulse la tecla de retroceso (backspace) una vez se inicie el salvapantallas.

Si utiliza la aplicación como salvapantallas debe tener en cuenta que en ciertos algoritmos se utilizan bastantes recursos, por tanto si mueve el ratón o pulsa una tecla, en ocasiones puede tardar varios segundos en desactivarse el salvapantallas.

Particularidades de la implementación:

El principal objetivo de la aplicación es establecer comparaciones entre diversos algoritmos de cierre convexo.

El método gráfico es el más intuitivo para que el usuario de la aplicación juzgue la capacidad de cada algoritmo al verlos ejecutar simultáneamente, pero esta comparación debe estar apoyada por algún resultado numérico que permita evaluar cuantitativamente la diferencia de tiempo/operaciones.

Tras analizar estos requisitos básicos es fácil llegar a la conclusión de que si se dispone de unos algoritmos que puedan ejecutar operaciones de una en una podremos sincronizarlos gráficamente y también contabilizar cuantas son necesarias para hallar el cierre convexo.

La implementación de los algoritmos en realidad no ha sido modificada, solo se ha alterado en la forma, no en el resultado ni en el número de operaciones empleadas.

La solución para permitir que estos algoritmos ejecuten paso a paso ha sido crear una estructura de datos que almacene la situación del algoritmo y permita así reconocer cual fue la última operación. (El problema de esta solución es que para cada algoritmo distinto es necesario utilizar un tipo de datos diferente.)

Notas sobre la implementación de cada algoritmo:

La implementación del algoritmo de **Fuerza Bruta** (Búsqueda de aristas) es la más sencilla de todas. Este es un método que aparece como primera idea de quien se enfrenta al problema de resolver un cierre convexo, y por lo tanto su desarrollo es el más intuitivo.

No requiere mantener estructuras de datos complejas y se puede realizar en muy pocas líneas de código. (Sin embargo es el que peor complejidad tiene).

El algoritmo de **Jarvis** es una modificación, una mejora, del algoritmo de búsqueda de aristas. La mejor elección de los puntos permite a este algoritmo mejorar en un orden respecto de su predecesor.

A efectos de código se puede reutilizar perfectamente el del algoritmo de búsqueda de aristas y la implementación se reduce a alterar las líneas correspondientes a la elección de los puntos.

De los cuatro algoritmos elegidos para la aplicación, el mayor reto se encuentra en la implementación del algoritmo **Incremental**, ya que la cantidad de funciones a programar es bastante significativa. (A continuación se enumeran y describen brevemente)

- Encontrar un punto dentro del cierre convexo.

El proceso de resolución del algoritmo incremental necesitamos iniciarlo tomando un punto dentro del cierre convexo. Para asegurar esta condición se toma el punto central a los tres primeros puntos: (Tomar de los tres puntos la mayor y menor ordenada y abscisa y calcular el punto medio.)

- Determinar si dado un punto P y un polígono convexo Q, el punto esta contenido en el polígono.

Posteriormente es necesario determinar para los siguientes puntos si se encuentran dentro o fuera del polígono formado por el cierre convexo (hasta ese momento). Para ello es necesario verificar si todas las aristas del polígono convexo Q dejan a su izquierda al punto P. (Sentido positivo)

- Calcular las rectas soporte de un punto respecto de un polígono convexo.

Si se encuentra un punto fuera del polígono se deberá añadir al cierre convexo, para ello se deben conocer sus rectas soporte.

Para calcular la recta soporte superior se une el punto P con un punto cualquiera de Q. (En la implementación elegida se inicia con el punto medio del polígono). Y, se comprueba si la recta deja a todos los puntos del polígono a la izquierda. En caso afirmativo se ha encontrado la recta soporte superior. En caso contrario es necesario continuar probando. El siguiente punto en ser evaluado dependerá del ángulo que forme la recta con el polígono. (Según si es cóncavo o convexo avanzaremos o retrocederemos en el polígono).

Para calcular la recta soporte inferior se procede de forma análoga. (Pero buscando que todos los puntos queden a la derecha)

- Eliminar e incluir puntos en la estructura de datos del cierre convexo en tiempo constante.

Una vez determinadas las rectas soporte, se eliminan los puntos del cierre convexo contenidos entre los soportes y se inserta el nuevo punto.

Realizar la inserción y eliminación de puntos en tiempo constante requiere una estructura bastante compleja (un árbol equilibrado o semejante).

En la aplicación no se ha utilizado ningún tipo de estructura especial para manejarlo, y se requieren bastantes instrucciones para efectuar las inserciones y eliminaciones. Pero dado el carácter particular de la simulación nos es posible en un solo paso (aunque en varias instrucciones) insertar o eliminar un punto de la estructura del cierre convexo. Así, a efectos prácticos simularemos el algoritmo correctamente, con inserciones y eliminaciones en tiempo constante.

El algoritmo de **Graham** requiere una ordenación previa. Esta ordenación se ha incluido en la simulación.

Realmente se realiza toda la fase de ordenación en conjunto, y posteriormente se simula una espera correspondiente al número de operaciones que han sido necesarias para realizar la ordenación (implementada mediante un algoritmo de quicksort).

Así pues, el algoritmo de Graham solo tiene que comenzar a procesar con todos los puntos ya ordenados angularmente alrededor del primer punto que no perteneciese al cierre convexo.

En las simulaciones de la aplicación, se muestra el tiempo necesario para realizar la ordenación con el fin permitir unas comparaciones realistas con los métodos sin pre-proceso.

A tener en cuenta:

El número de operaciones realizadas por cada algoritmo para hallar el cierre convexo no es totalmente significativo. Debe tomarse como una medida de orientación.

El código puede ser optimizado para reducir este número de operaciones. Lo que es realmente significativo y lo que se pretende contrastar es el orden de complejidad obtenido por los algoritmos, que es independiente de la optimización del código.