

Tarea 2 - Cerradura Convexa

Profesora: Nancy Hitschfeld Kahler

Auxiliar: Sergio Salinas

Ayudante: Andrés Abarca

1. Introducción

La cerradura convexa (ver Figura 1), aparte de ser un conjunto matemático de puntos que cumplen ciertas condiciones, es algo que va más allá de la teoría y se utiliza activamente en distintas aplicaciones. Tales como reconocimiento de patrones, procesamiento de imágenes, información geográfica e incluso cosas como la teoría de juegos.

Por todo ello y mucho más, hoy en día existen al menos 8 formas distintas de implementar dicho algoritmo y el objetivo principal de esta tarea, será probar al menos dos de ellas utilizando como base su Tarea 1.

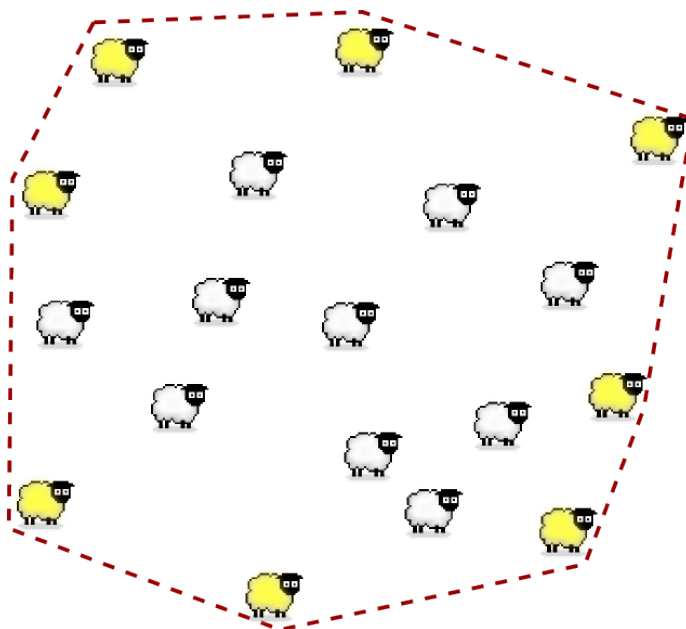


Figura 1: Ejemplo de una cerradura convexa compuesta por 8 lindas ovejitas.

1. Elementos a implementar

Esta tarea será implementada sobre su Tarea 1 y, por tanto, toda debe estar desarrollada en C++. La tarea está dividida en tres secciones:

1. Debe usted implementar el algoritmo de Gift Wrapping, el cual debe recibir como argumento un listado de puntos y debe retornar el polígono que representa la cerradura. Puede usted guiarse por la siguiente firma: `Polygon giftWrapping(Point[] cloud)`. La clase punto y la clase polígono debe ser la misma que la utilizada en la Tarea 1.
2. Como Gift Wrapping tiene peor caso $O(n^2)$ y podemos estar trabajando con nubes de millones de puntos, es necesario poder utilizar otro algoritmo de cerradura convexa cuya complejidad computacional es $O(n \log n)$. Por ello, usted debe implementar UNO de los siguientes algoritmos (que siga la misma firma que Gift Wrapping):
 - Incremental (ordena, construye el upper-hull y el lower-hull)
 - Graham scan
 - Divide and conquer
3. Una vez implementado ambos métodos, diseñe un experimento (ver lectura 2) para determinar para qué distribuciones de puntos es mejor usar uno u otro. Por ejemplo, ¿Cuál es mejor con puntos generados aleatoriamente? ¿Cuál es mejor si un porcentaje pequeño de puntos pertenece a la cerradura convexa? ¿Se puede determinar ese porcentaje? Implemente además un método que genere una lista aleatoria de n puntos, y otro que genere un conjunto de puntos en que un cierto porcentaje de estos forme parte de la cerradura convexa y el resto se distribuya aleatoriamente en el interior. Estos métodos construyen la lista de puntos con que se llamarán a continuación los algoritmos de cerradura convexa. Experimente con distintos tamaños de n , por ejemplo $n = 10^4, 10^5, 10^6$ y con distinto porcentaje de puntos formando parte de la cerradura. ¿Cómo se desempeñan los algoritmos? ¿Existen diferencias en cuanto a tiempo de cálculo? OJO: También debe comparar que una vez terminado el procesamiento de ambas cerraduras, estas deben ser idénticas, ya que la cerradura convexa es única.

2. Información extra

- Para esta tarea se debe incluir **CMake** y **Gtest**, se recomienda usar la plantilla de proyecto de C++ vista en las auxiliares.
- Los resultados de sus comparaciones además de estar en testing, deben venir en un archivo `.pdf` llamado `ANALISIS.pdf`, donde debe indicar si tomó algún supuesto en

las implementaciones y bajo qué condiciones obtuvo sus resultados, además de estos últimos.

- Se pide además graficar el tiempo obtenido para distintos valores de n como una forma de verificar que su implementación sigue el comportamiento esperado, e incluir los gráficos en el archivo **ANALISIS.pdf**.
- La fecha de entrega así como los contenidos de la tarea podrían llegar a modificarse según las necesidades del curso y cómo este avance, pero de ocurrir esto se notificará en clases y posteriormente se verá reflejado en U-Cursos.
- Debe incluir README sobre cómo ejecutar su programa.