Computación Geométrica
Tema 1: Introducción a la GC

Introducción a la Geometría Computacional

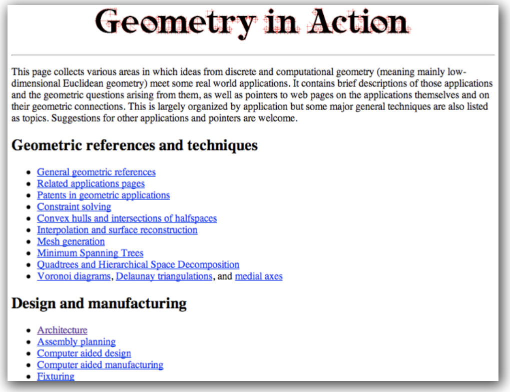
Índice

Intro Geometría Computacional

Definición

- **Definición: Estudio de algoritmos y estructuras de datos eficientes para problemas geométricos.**
- **Surge a mitad de la década de los 70 (tesis de Shamos)**
- **En la actualidad:**
  - **Interés en trabajos prácticos y aplicados**
  - **Desarrollo científico estable**

Áreas de aplicación



Geometry in Action

This page collects various areas in which ideas from discrete and computational geometry (meaning mainly low-dimensional Euclidean geometry) meet some real world applications. It contains brief descriptions of those applications and the geometric questions arising from them, as well as pointers to web pages on the applications themselves and on their geometric connections. This is largely organized by application but some major general techniques are also listed as topics. Suggestions for other applications and pointers are welcome.

Geometric references and techniques

- General geometric references
- Related applications pages
- Patents in geometric applications
- Constraint solving
- Convex hulls and intersections of halfspaces
- Interpolation and surface reconstruction
- Mesh generation
- Minimum Spanning Trees
- Quadtrees and Hierarchical Space Decomposition
- Voronoi diagrams, Delaunay triangulations, and medial axes

Design and manufacturing

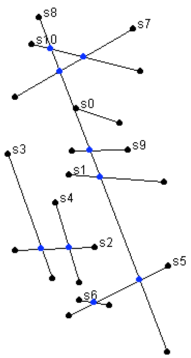
- Architecture
- Assembly planning
- Computer aided design
- Computer aided manufacturing
- Fixturing

http://www.ics.uci.edu/~eppstein/geom.html

Campo teórico-práctico

- **Teoría:**
  - **Cálculo de costes de algoritmos**
  - **Encontrar algoritmos óptimos**
  - **Demostraciones formales**
- **Práctica**
  - **Desarrollo de aplicaciones y librerías**
  - **Tratamiento de casos degenerados**

Intersección de segmentos



Recubrimiento Convexo

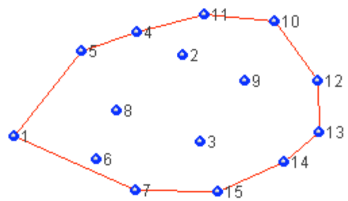
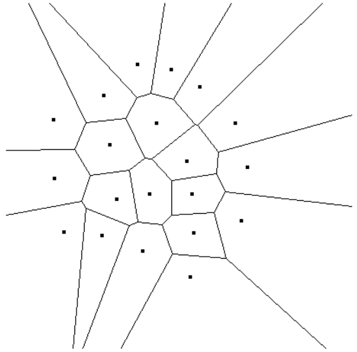
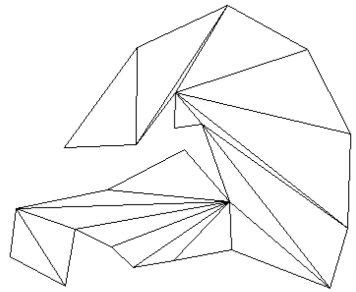


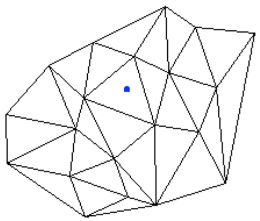
Diagrama de Voronoi



Triangulación de polígonos



Localización de puntos



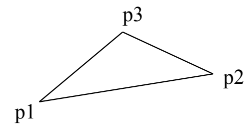
Ver ejemplos de applets

- **Uno de los objetivos del curso es que, cuando termines, seas capaz de implementar un applet en Java con alguno de estos algoritmos**
- **Algunos ejemplos de otros años**
  - **Intersección de segmentos**
  - **Triangulación Delaunay y Diagrama de Voronoi**
  - **Triangulación de polígonos**

Intro Geometría Computacional

- Definición y problemáticas de la Geometría Computacional
- Ejemplos de algoritmos geométricos
  - Sentido de giro de un triángulo
  - Punto a derecha o izquierda
  - Punto dentro de un triángulo
  - Convexidad de vértices de polígonos
  - Área de un polígono
  - Recubrimiento convexo

Sentido de giro de un triángulo (1)



- **Podemos calcular el área (con signo) del triángulo a partir del producto vectorial de los vectores (p1,p2) y (p1,p3)**

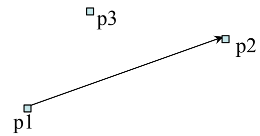
Sentido de giro de un triángulo (2)

- **Determinante:**

$$\Delta(p_1, p_2, p_3) = \frac{1}{2} \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}$$

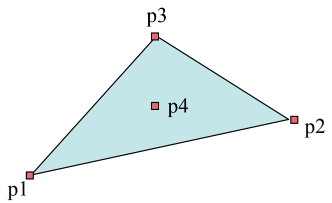
- **Signo del determinante:**
  - **>0 cuando a,b y c forman un "giro antihorario"**
  - **<0 cuando a, b y c dibujan un "giro horario"**
- **Ejemplo: p1 = (1,1); p2 = (1,3); p3 = (3,1)**

Punto a derecha o izquierda



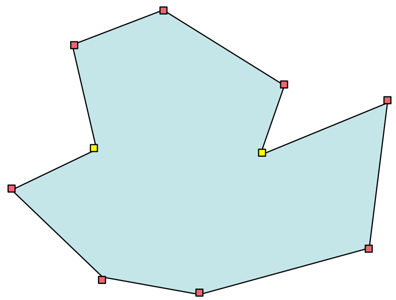
- **El cálculo del signo del área del triángulo (p1,p2,p3) se aplica con facilidad a determinar si p3 se encuentra a la dcha. o izqda. del segmento orientado (p1,p2).**

Punto dentro de un triángulo



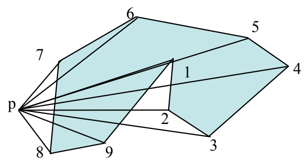
¿Cómo saber si un punto p4 está dentro o fuera del Triángulo formado por (p1,p2,p3)?

Convexidad de vértices de un polígono



¿Cómo saber si un vértice de un polígono es cóncavo o convexo?

Área de un polígono

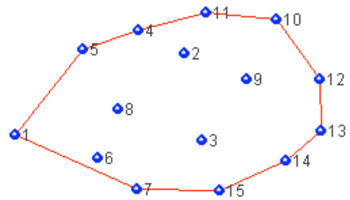


- **p es un punto cualquiera del plano**
- **El área del polígono se puede expresar por**

$$A(P) = A(p, v_0, v_1) + A(p, v_1, v_2) + \ldots + A(p, v_{n-1}, v_0)$$

Más resumida: $$2A(T) = \sum_{i=0}^{n-1} (x_i y_{i+1} - y_i x_{i+1})$$

Convex hull



- **Propiedad: una arista es parte del convex hull de un conjunto de puntos S si y sólo si deja todos los puntos del conjunto a su izquierda**

Algoritmo del convex hull

Para cada p_i hacer
Para cada $p_j \neq p_i$ hacer
    Para cada $p_k \neq p_i \neq p_j$ hacer
        Si p_k no está a la izqda de (p_i,p_j)
            entonces (p_i,p_j) no es extremo

- **Coste del algoritmo: $O(n^3)$**

A considerar en algoritmos de GC

- **Entender la formulación geométrica del problema**
- **Casos degenerados**
- **Robustez frente a errores de redondeo**
- **Eficiencia (utilizar las estructuras de datos aducadas)**

Medidas de la eficiencia

- **Tiempo de respuesta**
- **Coste de almacenamiento**
- **Tiempo de preprocesamiento**
- **Tiempo de actualización de una estructura**

