TRIANGULACIONES

PARTE I: TRIANGULACIÓN DE UNA NUBE DE PUNTOS

Definición: *Dado un conjunto $S=\{s_1, ..., s_n\}$ de puntos de R^2, llamamos triangulación de*
de su cierre convexo en triángulos, tal que cada punto de S sea un vértice de un triángul

Consideremos el siguiente algoritmo:

Entrada: **$S=\{s_1, ..., s_n\}$**

1. **Para cada par de vértices, añadir el segmento que los une si no corta a ningún anterior.**

Este algoritmo voraz calcula una triangulación de S, con coste en tiempo $O(n^2)$. Sin emba triangulación que proporciona dicho algoritmo no es útil en la mayoría de las aplicaciones ejemplo en la interpolación de una función continua sobre una superficie.

Esta función puede representar un mapa de alturas, que se miden sólo en ciertos puntos debe ser interpolada en el resto (esto es frecuente en los GIS). Podría darse una situació siguiente:

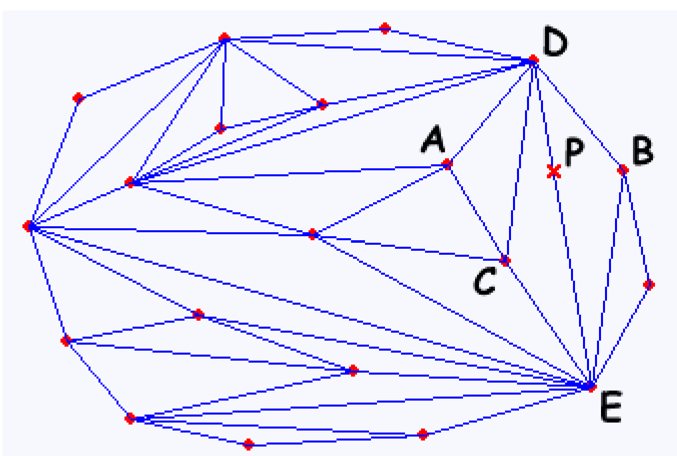


FIGURA 1

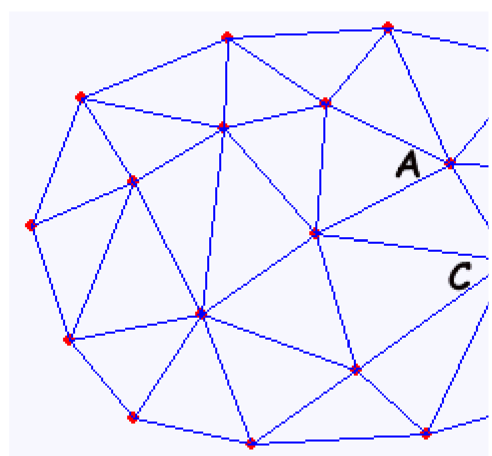


FIGURA 2

En la **figura 1** *se calcula la altura de P interpolando las de C, D, E. Si las alturas medidas son 100m, 70m y 20m respectivamente, la altura en P resultaría 53m. Si las alturas en A 110m respectivamente, este valor para P resulta inadecuado. Esto no ocurre en la triangu* **figura 2***, en la que el valor en P depende de A, B y C, obteniéndose una altura de 106m.*

Definición: *Se dice que una triangulación T_1 es mejor que otra T_2, y se representa T_2<=*
(en orden lexicográfico), donde A_{T1} y A_{T2} son las listas de los ángulos ordenados de mer
T_1 y T_2 respectivamente.

Definición: *Una triangulación T se dice equilátera si no existe otra triangulación T' tal qu*

La **figura 2** *es un ejemplo de triangulación equilátera, mientras que la de la* **figura 1** *no lc*

Una triangulación equilátera reduce el número de ángulos muy agudos, por lo que es me
que requieran precisión. Por esto es frecuente que en muchas aplicaciones sea preferibl
triangulación de este tipo a otra cualquiera.

La triangulación de una nube de puntos guarda una estrecha relación con el diagrama de
dicha nube de puntos. Así, se tiene la siguiente

Definición: *Dado un conjunto $S=\{s_1, ..., s_n\}$ de puntos de R^2, el dual geométrico de su di*
Voronoi es una triangulación de los puntos de S. Dicha triangulación se llama triangulacic

Es decir, la triangulación de Delaunay se obtiene uniendo vecinos de Voronoi. Posee vari
importantes:

- *Minimiza el máximo radio de una circunferencia circunscrita.*
- *Maximiza el mínimo ángulo (de hecho, la triangulación de Delaunay es equilátera).*
- *Maximiza la suma de los radios de las circunferencias inscritas.*
- *La distancia entre cualquier par de vértices a través de aristas de la triangulación es constante (2'42) por su distancia euclídea.*

Sin embargo, calcular la triangulación de Delaunay a través del diagrama de Voronoi es c
que el método sea óptimo, con coste O(n log n)), por lo que conviene disponer de otras te
conseguirla.

Uno de estos métodos consiste en usar la siguiente relación:

Proposición: *Sea $S=\{s_1, ..., s_n\}$ un conjunto de puntos de R^2. Si proyectamos S sobre ur*
con ecuación $z=x^2+ y^2$ (esto es, asociamos a cada punto $s_i=(xi, y_i)$ del plano el punto $(x_i,$
R^3), entonces la proyección sobre R^2 de la parte inferior del casco convexo de los puntos
paraboloide es la triangulación de Delaunay de S.

Así, el algoritmo sería el siguiente

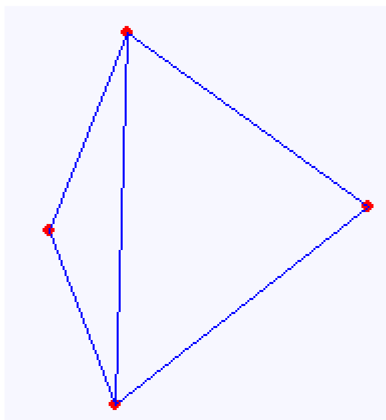
Entrada: **$S=\{s_1, ..., s_n\}$**

1. **Proyectar los puntos de S sobre el paraboloide $z=x^2+ y^2$.**
2. **Calcular el casco convexo de los nuevos puntos.**
3. **Proyectar la parte inferior sobre el plano.**

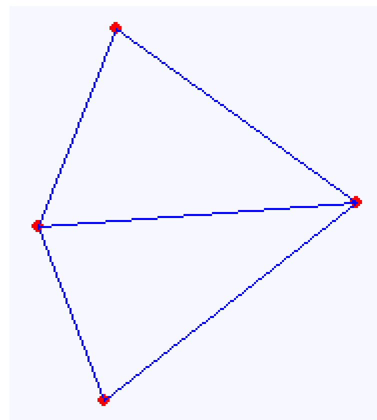
Un método más fácil de calcular la triangulación de Delaunay, aunque con un coste O(n^2, giros de aristas, modificando una triangulación inicial hasta llegar a la de Delaunay.

Definición: *Dados dos triángulos que comparten un lado, formando un cuadrilátero conv el lado común es legal si maximiza el ángulo mínimo. El lado común es ilegal en caso co*

Definición: *Dados dos triángulos que comparten un lado, formando un cuadrilátero conv denomina flip (giro) en la diagonal común al cambio de dicha diagonal por la otra en el cu dice que un flip es positivo si convierte una diagonal ilegal en legal.*



→ *Flip* →



Ilegal *Legal*

Teorema: *Una triangulación es de Delaunay si y sólo si no contiene aristas ilegales.*

Además, para saber si una diagonal es legal basta comprobar si la circunferencia que pa puntos de la misma y un punto cualquiera del cuadrilátero al que pertenece no contiene a caso contrario, la diagonal es ilegal.

Por tanto, un algoritmo para obtener la triangulación de Delaunay es el siguiente:

Entrada: **T, triangulación inicial cualquiera**

1. **Poner todas las aristas internas en una cola**
2. **Mientras la cola no esté vacía**
   1. **Sacar una arista, a, de la cola**
   2. **Si C_a tiene diagonal ilegal, hacer un flip positivo y añadir las aristas exte la cola**

(C_a es el cuadrilátero con diagonal a)

APPLET 1

TRIANGULACIÓN DE DELAUNAY

Se puede disponer de una variante de este método, conocida como algoritmo incrementa puntos de la nube se introducen poco a poco, y la triangulación se optimiza tras cada ins

Entrada: **T, triangulación de Delaunay, P, nuevo punto, p_{-1}, p_{-2}, p_{-3}, triángulo que co**

1. **Encontrar el triángulo $p_i p_j p_k$ en T en el que se encuentra P**
2. **Si $p_i p_j p_k$ contiene a P**
   1. **Añadir las aristas de P a p_i, p_j, p_k**
   2. **Legalizar las aristas $p_i p_j$, $p_i p_k$, $p_j p_k$**
3. **Si no (está sobre una arista, $p_i p_j$ por ejemplo)**
   1. **(Sea p_l el otro punto del triángulo que contiene a $p_i p_j$) Añadir aristas de F**
   2. **Legalizar las aristas $p_i p_l$, $p_l p_j$, $p_j p_k$ $p_k p_i$**

(Legalizar la arista $p_i p_j$ cuando se introduce p:

1. **Si $p_i p_j$ es ilegal**
   1. **(Sea $p_i p_j p_k$ el triángulo adyacente a $pp_i p_j$) Reemplazar $p_i p_j$ por pp_k**
   2. **Legalizar $p_i p_k$, $p_j p_k$)**

Existen muchos criterios de optimización. Uno de los más famosos se conoce como **trian peso mínimo***, y consiste en buscar una triangulación de la nube de puntos que minimice de las aristas. No se conoce ningún algoritmo de coste polinomial que resuelva este prob si es* **NP-completo***. El algoritmo que proporciona la mejor aproximación, de Levcopoulos una solución con un factor multiplicativo constante de la longitud óptima. Otro criterio bus triangulación que minimice la máxima longitud de las aristas. Edelsbrunner y Tan demost triangulación contiene las aristas del árbol recubridor mínimo (Minimmum Spannig Tree, proporciona un algoritmo de coste polinomial: calcular el MST y después triangular los pc resultantes, usando programación dinámica.*

PARTE II: TRIANGULACIÓN DE POLÍGONOS

Definición: *Dado un polígono P, una triangulación de P es una subdivisión de su interior tal que los vértices del polígono sean vértices de los triángulos y viceversa. Se denomina los lados de los triángulos (de cualquier triangulación de P) que no son lados de P.*

La triangulación de un polígono es un problema fundamental en geometría computaciona ampliamente usada en la teselación de geometrías curvadas, como las descritas por spli Manocha, 1994). Tiene importancia como preprocesamiento en diversos algoritmos, com en el problema de la galería de arte:

Definición: *Decimos que un punto y es visible desde otro punto x en un polígono simple segmento que los une está completamente contenido en el interior de P.*

Problema de la galería de arte: *Dado un polígono simple P, que representa la planta de podemos considerar el siguiente problema: dónde poner 'guardias' para que cada punto vigilado por alguno de ellos.*

La solución a este problema, que se describe a continuación, pasa por calcular una triang planta del edificio.

Teorema (de la galería de arte): *Dado un polígono simple con n vértices, existe un conj que lo vigilan por completo con, a lo sumo,* $\lfloor n/3 \rfloor$ *guardias.*

Lema: *Sea T el grafo resultante de una triangulación de un polígono simple. Entonces T*

Entonces, es posible triangular la planta del edificio y colorearla usando sólo 3 colores, q aparecer en cada uno de los triángulos. Por tanto, si se toman todos los vértices que ten menos usado se vigila el edificio completo, ya que se vigila cada uno de los triángulos qu

Existen muchas aplicaciones en las que, al igual que el caso de nubes de puntos, es imp descomponer polígonos en triángulos con formas especiales (por ejemplo, evitando que muy agudos).

La triangulación restringida de Delaunay, proporciona un método para forzar la aparición un PSLG (Planar Straight-Line Graph), G, en la triangulación de Delaunay. Un triángulo a la triangulación restringida de Delaunay si su circunferencia circunscrita no contiene ni p vértice de G visible desde cualquiera de los puntos de abc. Esta definición generaliza la triangulación de Delaunay en el caso de que G no contenga aristas. Si G es un polígono, triangulación restringida de Delaunay contiene sólo triángulos interiores a G.

Sin embargo también existen otras muchas aplicaciones en las que la forma de los triáng indiferente. Sólo consideraremos el problema de, dado un polígono simple, calcular una t suya.

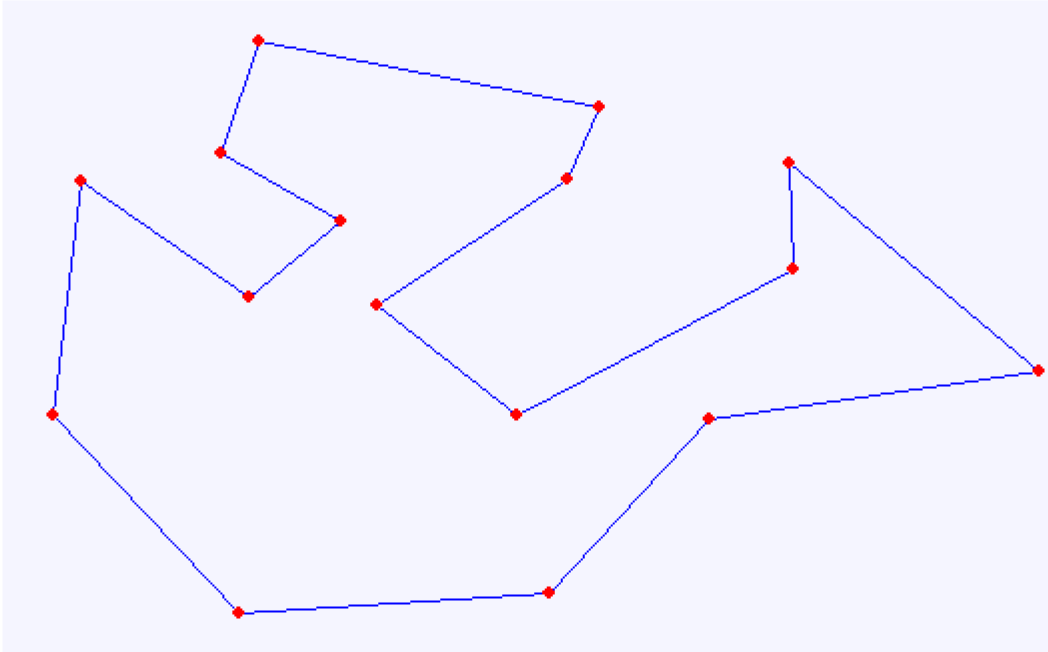
Es posible encontrar un algoritmo que resuelve este problema en tiempo polinomial, bas diagonales :

Lema: *Toda diagonal divide un polígono en dos, con menor número de vértices.*

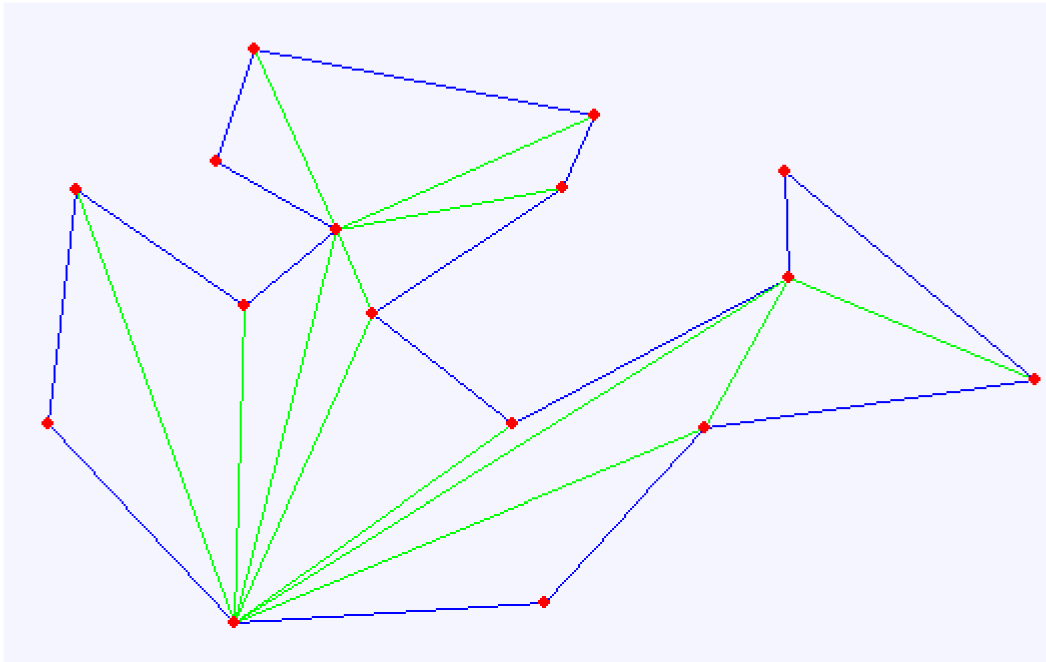
Lema: *Todo polígono con más de 3 vértices admite una diagonal.*

Teorema: *Todo polígono admite una triangulación.*

Esto proporciona un método para triangular polígonos con coste en tiempo $O(n^3)$*.*



ANTES



DESPUÉS

APPLET 2

TRIANGULACIÓN DE POLÍGONOS

Existen muchos algoritmos que resuelven el problema con coste en tiempo O(n·log n), pe *abierto durante mucho tiempo fue determinar si existe un algoritmo de coste O(n). Este p resuelto en 1991 por Chazelle, aunque el algoritmo resultó ser tan complicado que no pu con los algoritmos prácticos, aunque asintóticamente peores, de coste O(n·log n). El algo Chazelle reduce el problema de la triangulación de un polígono P al de calcular el mapa c horizontal de P, esto es, la partición del polígono obtenida trazando líneas horizontales a derecha de los vértices.*

Si se busca optimizar propiedades se puede recurrir a los siguientes algoritmos:

PROPIEDAD	ALGORITMO	ORDEN
Delaunay	*Varios*	*O(n·log n)*
Minimiza máximo ángulo	*Inserción rápida de aristas*	*O(n^2·log n)*
Minimiza máxima pendiente	*Inserción de aristas*	*O(n^3)*
Minimiza longitud total	*Algoritmos de aproximación*	*O(n·log n)*

Uno de los algoritmos de coste O(n·log n) se basa en la definición de polígono monótono considera el caso de polígonos monótonos, para luego generalizarlo a polígonos cualesq descomponiéndolos en una colección de polígonos monótonos disjuntos. La triangulación monótonos tiene lugar en tiempo O(n) (Fournier y Montuno, 1984).

Definición: *Una cadena poligonal, C, se dice estrictamente monótona con respecto a un si cualquier línea ortogonal a L intersecta a C en, a lo sumo, un punto. La cadena poligon monótona con respecto a L si toda línea ortogonal a L intersecta a C en un segmento úni*

Definición: *Un polígono, P, se dice monótono con respecto a una línea L, si su frontera p descomponerse en dos cadenas monótonas con respecto a L.*

El coste para comprobar si un polígono es monótono con respecto al eje horizontal es O(

Triangulación de polígonos monótonos:

El método que se describe a continuación es un método voraz.

Dado un polígono monótono P, el primer paso es buscar los vértices visibles para cada v Cada vez que se encuentra un vértice u visible desde v, se añade la arista que los une. A proceso se debe haber triangulado P (como se vio anteriormente). Este algoritmo se pue cualquier PSLG, sin embargo, su coste es muy elevado. Por otra parte, si el polígono es proceso anterior puede realizarse en tiempo O(n):

Algoritmo para la triangulación de un polígono monótono:

Entrada: **P, polígono monótono con n vértices**

1. **Ordenar los vértices de P según coordenada y decreciente, resultando la lista**

(n)}

2. **Guardar v(1) y v(2) en una pila, PILA. Sea i=3**
3. **(Sea {u(1), ... u(s)} el contenido de la PILA, donde la cima es u(s)) Si v(i) es ady pero no a u(s), entonces**
   1. **añadir las aristas {v(i), u(2)}, ..., {v(i), u(s)},**
   2. **sacar todos los vértices de la PILA,**
   3. **poner u(s) y v(i) en la PILA,**
   4. **i← i+1,**
   5. **ir al paso 6**
4. **Si v(i) es adyacente a u(s), pero no a u(1), entonces**
   1. **mientras el vértice bajo la cima de la PILA (sea u') sea visible desde v(i)**
      1. **añadir una arista {v(i), u'}**
      2. **sacar la cima de la PILA**
   2. **poner v(i) en la PILA**
   3. **i← i+1**
   4. **ir al paso 6**
5. **Si v(i) es adyacente a u(s) y a u(1), entonces**
   1. **añadir las aristas {v(i), u(2)}, ..., {v(i), u(s-1)}**
   2. **sacar todos los vértices de la PILA y parar**
6. **Si i<=n, volver al paso 3.**

Triangulación de un polígono simple cualquiera:

El problema ahora es encontrar una descomposición de un polígono simple en polígonos Para esto se emplea un algoritmo que realiza un barrido del plano. Se añadirán diagonale polígono en piezas monótonas.

La ausencia de monotonía (horizontal) tiene lugar sólo en los vértices cuyo ángulos interi 180º y ambas aristas están a su izquierda o a su derecha. La notación común para el prir vértice es vértice de unión (merge vertex), y para el segundo vértice de separación (split al sentido del barrido.

Al encontrar un vértice de separación (aristas a la derecha), existe una arista e_j superior y inferior. Se une entonces el vértice al más cercano a la izquierda de la línea de barrido qu e_k. A éste vértice se le denomina auxiliar(e_j). Si no hay ningún vértice entre estas aristas, está definido como el extremo izquierdo de una de las dos aristas que esté más cerca de barrido.

Los elementos básicos del algoritmo son:

- **Puntos de parada:** *Los extremos de los segmentos del polígono. Se ordenan por c de coordenada x.*
- **Estado de barrido:** *El estado de la línea de barrido viene dado por la lista de arista intersectan a la propia línea de barrido, ordenada de arriba abajo.*
- **Procesamiento de los puntos de parada:** *Hay seis tipos de puntos de parada. Se actual en el que está detenida la línea:*
  - **Vértice de separación:** *Se busca en el estado de la línea de barrido la arista inmediatamente superior a v. Se añade una diagonal uniendo v a auxiliar(e). S dos aristas incidentes en v al estado del barrido, y se pone v como el auxiliar c*

- *de las dos aristas, y como nuevo auxiliar de e.*
  - **Vértice de unión:** *Se encuentran las dos aristas incidentes a este vértice en e barrido (deben ser adyacentes). Se eliminan ambas (de la lista). Sea e la arist inmediatamente superior a ambas. Poner v como nuevo auxiliar de e.*
  - **Vértice de comienzo:** *(Ambas aristas están a la derecha de v, con ángulo int 180º) Se inserta este vértice y sus aristas en el estado de barrido.*
  - **Vértice final:** *(Ambas aristas están a la izquierda de v, con ángulo interior me eliminan ambas aristas del estado de barrido.*
  - **Vértice de la cadena superior:** *(Una arista está a la izquierda, y la otra a la c interior del polígono está debajo) Se cambia la arista izquierda por la arista de estado de barrido. Se pone v como el auxiliar de la nueva arista.*
  - **Vértice de la cadena inferior:** *(Una arista está a la izquierda, y la otra a la de interior del polígono queda encima del vértice) Se cambia la arista izquierda p derecha en el estado de barrido. Sea e la arista que pasa por encima de v. Se auxiliar de e.*

Esto sólo inserta arista para los vértices de separación. Para procesar los vértices de uni algoritmo esencialmente igual que realice el barrido de derecha a izquierda. Esto podría i diagonales repetidas, aunque este problema puede solucionarse con cierta atención espe cambios de vértice auxiliar. Existen muchos casos especiales, que pueden ser tratados fa que el algoritmo es muy eficiente.

Los métodos vistos hasta ahora describen algoritmos deterministas para resolver el probl triangulación. Sin embargo, es posible usar un método no determinista. Esto es lo que ha incremental no determinista de Seidel, cuya complejidad esperada es O(n·log n). En la p tiempo empleado por este algoritmo en la triangulación de un polígono simple es casi line algoritmo consta de tres partes:*

1. *Descomponer el polígono en trapezoides.*
   Sea S un conjunto de segmentos (no horizontales) del polígono que no intersectan algoritmo no determinista se usa para crear la descomposición trapezoidal del plano segmentos de S. Esto se hace tomando una ordenación cualquiera s1, ..., sN de los S y añadiendo un segmento de cada vez para construir los trapezoides. La restricci haya segmentos horizontales se impone para limitar el número de vecinos de cada embargo, no hay pérdida de generalidad). El número de trapezoides es lineal con re número de segmentos. Seidel demuestra que, si cada permutación de s1, ..., sN es probable, la construcción de los trapezoides lleva un tiempo esperado O(n·log n).*
2. *Descomponer los trapezoides en polígonos monótonos.*
   Esta operación tiene un coste lineal.
3. *Triangular los polígonos monótonos.*
   Recuérdese que esto puede hacerse en tiempo lineal.

BIBLIOGRAFÍA:

- *"Fast Polygon Triangulation based on Seidel's Algorithm", Atul Narkhede, Dinesh M Department of Computer Science, UNC Chapel Hill.*
- *"Computational Geometry in C (Second Edition)", Joseph O'Rourke (capítulo 22, Tr*
- *"Computational Geometry, Methods and Applications", Jianer Chen; Computer Scie Texas A&M University.*

- *"Computer Graphics (Lecture Notes)", (Spring 1997) Dave Mount.*
- *"TEMA 6: Diagramas y triangulaciones", Domingo Gallardo; DCCIA, Universidad de*
- *"Sesión 6: Triangulación de polígonos", Domingo Gallardo; DCCIA, Universidad de*

OTRAS REFERENCIAS:

- *"A polynomial time algorithm for the minmax angle triangulation", H. Edelsbrunner, T Waupotitsch; SIAM J. Sci. Statist. Comput.*
- *"Triangulating simple polygons and equivalent problems", A. Fournier y D.Y. Montur on Graphics.*
- *"Incremental Delaunay triangulation", Dani Lischinski; Academic Press, Boston.*

Auto
Manue
Je