

# Triangulaciones de Delaunay

Dr. Eduardo A. RODRÍGUEZ TELLO

CINVESTAV-Tamaulipas

22 de marzo del 2013



Cinvestav

- 1 Triangulaciones de Delaunay
  - Introducción
  - Árbol generador mínimo
  - Propiedades de los generadores geométricos
  - Maximización de ángulos e intercambio de aristas
  
- 2 Algoritmo de construcción incremental
  - Introducción
  - Prueba en círculo
  - Actualización incremental
  - Localización de un punto
  - Cambios estructurales
  - Reagrupamiento de sitios



- El material de esta clase está basado en el capítulo 9 del libro: Mark de Berg, Otfried Cheong, Marc van Kreveld and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition (April 16, 2008), ISBN-10: 3540779736.



- 1 Triangulaciones de Delaunay
  - **Introducción**
  - Árbol generador mínimo
  - Propiedades de los generadores geométricos
  - Maximización de ángulos e intercambio de aristas
- 2 Algoritmo de construcción incremental



# Introducción

- Hoy vamos a considerar una estructura relacionada con los diagramas de Voronoi llamada Triangulación de Delaunay (TD)
- Recordemos que el diagrama de Voronoi de un conjunto de sitios en el plano es una subdivisión planar
- El dual de dicha subdivisión es otra subdivisión que se define de la siguiente manera



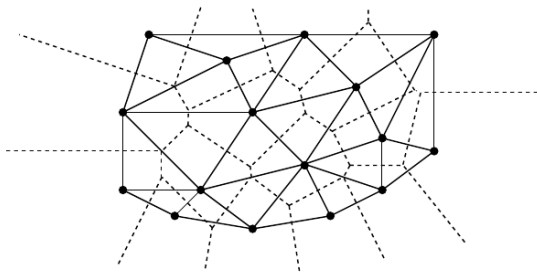
# Introducción

- Para cada cara del diagrama de Voronoi, se crea un vértice (correspondiente al sitio)
- Para cada arista del diagrama de Voronoi que cae entre dos sitios  $p_i$  y  $p_j$ , se crea una arista en el dual conectando estos dos vértices
- Finalmente, cada vértice del diagrama de Voronoi corresponde a una cara del dual
- El grafo dual resultante es una subdivisión planar



# Introducción

- Asumiendo la posición general, los vértices del diagrama de Voronoi (líneas punteadas) tienen grado tres, por lo tanto las caras del grafo dual resultante (excluyendo la cara exterior) son triángulos
- Así el grafo dual resultante es una triangulación de los sitios, llamada **triangulación de Delaunay** (líneas sólidas)



# Introducción

- Las triangulaciones de Delaunay tienen una serie de propiedades interesantes, que son consecuencias de la estructura del diagrama de Voronoi
- **Cubierta convexa:** La frontera de la cara exterior de la triangulación de Delaunay es la frontera de la cubierta convexa del conjunto de puntos





# Introducción

- **Circunferencia circunscrita:** La circunferencia circunscrita de cualquier triángulo en la triangulación de Delaunay está vacía (no contiene ningún sitio de  $P$ )

*Demostración:* Esto es por que el centro de este círculo es el correspondiente vértice de Voronoi dual, y por la definición del diagrama de Voronoi, los tres sitios que definen este vértice son sus vecinos más cercanos



# Introducción

- **Círculo vacío:** Dos sitios  $p_i$  y  $p_j$  están conectados por una arista en la triangulación de Delaunay, si y sólo si hay un círculo vacío que pasa por  $p_i$  y  $p_j$

*Demostración:* Si dos sitios  $p_i$  y  $p_j$  son vecinos en la triangulación de Delaunay, entonces sus celdas son vecinas en el diagrama de Voronoi, y por lo tanto para cualquier punto en la arista de Voronoi entre estos sitios, un círculo centrado en este punto pasando por  $p_i$  y  $p_j$  no puede contener cualquier otro punto. Por lo tanto las celdas de Voronoi de dos sitios están adyacentes en el diagrama de Voronoi diagrama, implicando que su arista esté en la triangulación de Delaunay



# Introducción

- **Par más cercano:** Los pares más cercanos de sitios en  $P$  son vecinos en la triangulación de Delaunay

*Demostración:* Supongamos que  $p_i$  y  $p_j$  son los sitios más cercanos. El círculo que tiene a  $p_i$  y  $p_j$  como su diámetro no puede contener ningún otro sitio, debido a que tal sitio estaría entonces más cerca de alguno de esos dos puntos, violando la hipótesis de que estos puntos son el par más cercano. Por lo tanto, el centro de este círculo está en la arista de Voronoi entre estos puntos, y de esta forma éste es un círculo vacío



# Introducción

- Si los sitios no están en la posición general, en el sentido de que cuatro o más sean cocirculares, entonces la triangulación de Delaunay puede no ser una triangulación, sino sólo un grafo planar
- Esto debido a que el vértice de Voronoi que es incidente a cuatro o más celdas de Voronoi inducirá una cara cuyo grado es igual al número de tales celdas
- En este caso el término más apropiado es **grafo de Delaunay**



# Introducción

- Sin embargo, es común asumir que los sitios están en la posición general o simplemente triangular las caras de grado cuatro o mayor en alguna forma arbitraria
- A partir de este momento asumiremos que los sitios están en la posición general
- Dado un conjunto  $P$  con  $n$  sitios donde hay  $h$  de ellos en la cubierta convexa, no es difícil demostrar por la fórmula de Euler que la triangulación de Delaunay tiene  $2n - 2 - h$  triángulos, y  $3n - 3 - h$  aristas



# Introducción

- La posibilidad de determinar el número de triángulos a partir de  $n$  y  $h$  sólo funciona en el plano
- En el 3-espacio, el número de tetraedros en la triangulación de Delaunay puede variar de  $O(n)$  hasta  $O(n^2)$
- En  $n$  dimensiones, el número de simplíces (la generalización  $d$ -dimensional de un triángulo) puede variar hasta  $O(n^{\lceil d/2 \rceil})$



- 1 Triangulaciones de Delaunay
  - Introducción
  - **Árbol generador mínimo**
  - Propiedades de los generadores geométricos
  - Maximización de ángulos e intercambio de aristas
- 2 Algoritmo de construcción incremental



# Árbol generador mínimo

- La triangulación de Delaunay posee algunas propiedades interesantes que no están relacionadas directamente con la estructura del diagrama de Voronoi
- Una de ellas es su relación con el **árbol generador mínimo** (*minimum spanning tree*)





# Árbol generador mínimo

- Dado un conjunto de  $n$  puntos en el plano, podemos pensar que los puntos definen un *grafo Euclidiano* cuyas aristas son todos los  $\binom{n}{2}$  pares (no dirigidos) de puntos distintos
- Cada arista  $(p_i, p_j)$  tiene un peso igual a la distancia Euclidiana entre  $p_i$  y  $p_j$
- Un árbol generador mínimo (AGM) es un conjunto de  $n - 1$  aristas que conectan los puntos (en un árbol libre) de forma que el peso total de las aristas sea mínimo
- Podemos calcular el AGM usando el algoritmo de Kruskal



# Árbol generador mínimo

- Recordemos que el algoritmo de Kruskal inicia ordenando las aristas e insertandolas una a una
- Podemos primero construir el grafo Euclidiano, y después pasar el resultado al algoritmo de Kruskal, para un tiempo total de ejecución  $O(n^2 \log n)$
- Sin embargo existe un método mucho más rápido basado en triangulaciones de Delaunay



# Árbol generador mínimo

- Primero se construye la triangulación de Delaunay del conjunto de puntos
- Más adelante veremos que esto puede hacerse en tiempo  $O(n \log n)$
- Posteriormente se calcula el AGM de la triangulación de Delaunay usando el algoritmo de Kruskal y se regresa el resultado
- Esto para un tiempo total de ejecución  $O(n \log n)$
- La razón por la que esto funciona está dada por el siguiente teorema



# Árbol generador mínimo

## Teorema

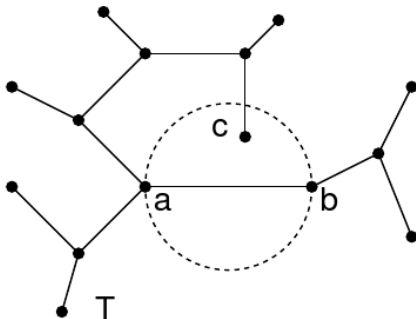
- El árbol generador mínimo de un conjunto de puntos  $P$  (en cualquier dimensión) es un subgrafo de la triangulación de Delaunay

## *Demostración*

- Sea  $T$  el AGM para  $P$  y  $w(T)$  el peso total de  $T$
- Sean  $a$  y  $b$  dos sitios cualesquiera tales que  $ab$  es una arista de  $T$
- Suponga por el contrario que  $ab$  no es una arista de la triangulación de Delaunay
- Esto implica que no existe un círculo vacío que pase por  $a$  y  $b$ , y en particular, el círculo cuyo diámetro es el segmento  $ab$  contiene un sitio, llamemoslo  $c$



# Árbol generador mínimo

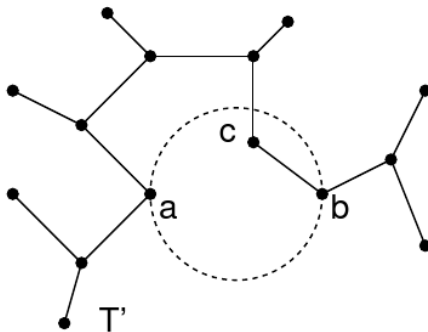


# Árbol generador mínimo

- La eliminación de  $ab$  del AGM divide el árbol en dos subárboles
- Asumiremos sin pérdida de generalidad que  $c$  cae en el mismo subárbol que  $a$
- Ahora, se remueve la arista  $ab$  del AGM y se agrega la arista  $bc$  en su lugar



# Árbol generador mínimo



# Árbol generador mínimo

- El resultado será un árbol generador  $T'$  cuyo peso es

$$w(T') = w(T) + \|bc\| - \|ab\| < w(T) \quad (1)$$

- La última desigualdad se da por que  $ab$  es el diámetro del círculo, implicando que  $\|bc\| < \|ab\|$
- Esto contradice la hipótesis de que  $T$  es el AGM, completando la demostración





- 1 **Triangulaciones de Delaunay**
  - Introducción
  - Árbol generador mínimo
  - **Propiedades de los generadores geométricos**
  - Maximización de ángulos e intercambio de aristas
- 2 Algoritmo de construcción incremental



# Propiedades de los generadores geométricos

- Una observación natural acerca de las triangulaciones de Delaunay es que sus aristas asemejan una red de transporte entre los puntos
- Al inspeccionar algunos ejemplos, es natural cuestionarse si la longitud del camino más corto entre dos puntos en una triangulación de Delaunay planar no es significativamente más larga que la distancia en línea recta entre esos puntos
- Esto está estrechamente relacionado con la teoría de generadores (*spanners*) geométricos, i.e. grafos geométricos cuyos caminos más cortos no son demasiado grandes



# Propiedades de los generadores geométricos

- Considere cualquier conjunto de puntos  $P$  y un grafo  $G$  cuyos vértices son los puntos en  $P$  y sus aristas son líneas rectas
- Para cualquier par de puntos  $p, q \in P$ , denotemos como  $\delta_G(p, q)$  la longitud del camino más corto entre  $p$  y  $q$  en  $G$ , donde el peso de cada arista es su distancia Euclidiana
- Dado un parámetro  $t \leq 1$ , decimos que  $G$  es un  $t$ -generador ( $t$ -spanner) si para cualquier par de puntos  $p, q \in P$ , el camino más corto entre  $p$  y  $q$  en  $G$  es a lo máximo un factor  $t$  más largo que la distancia Euclidiana entre estos puntos

$$\delta_G(p, q) \leq t \|pq\| \quad (2)$$



# Propiedades de los generadores geométricos

- Observe que cuando  $t = 1$ , el grafo  $G$  debe ser el grafo completo, que consiste de  $\binom{n}{2} = O(n^2)$  aristas
- Es muy interesante conocer si existen generadores con  $O(n)$  aristas
- Puede ser demostrado que las aristas de una triangulación de Delaunay forman un generador
- No demostraremos el siguiente resultado, el cual se debe a Keil y Gutwin
- **Teorema:** Dado un conjunto de puntos  $P$  en el plano, la triangulación de Delaunay de  $P$  es un  $t$ -generador para  $t = 4\pi\sqrt{3}/9 \approx 2.4$



- 1 Triangulaciones de Delaunay
  - Introducción
  - Árbol generador mínimo
  - Propiedades de los generadores geométricos
  - Maximización de ángulos e intercambio de aristas
- 2 Algoritmo de construcción incremental



# Maximización de ángulos e intercambio de aristas

- Otra interesante propiedad de las triangulaciones de Delaunay es que entre todas las posibles triangulaciones, la de Delaunay maximiza el ángulo mínimo
- Esta propiedad es importante, porque implica que las triangulaciones de Delaunay tienden a evitar triángulos demasiado delgados
- Esto es útil para muchas aplicaciones en las que los triángulos son usados para efectos de interpolación
- Además el teorema que veremos a continuación también se cumple



# Maximización de ángulos e intercambio de aristas

- Entre todas las triangulaciones con el mismo ángulo más pequeño, la triangulación de Delaunay maximiza el segundo ángulo más pequeño, y así sucesivamente
- En particular, cualquier triangulación puede estar asociada con una *secuencia ordenada de ángulos*, es decir, la secuencia incremental de ángulos  $(\alpha_1, \alpha_2, \dots, \alpha_m)$ , que aparecen en los triángulos que forman la triangulación
- Tenga en cuenta que la longitud de la secuencia será la misma para todas las triangulaciones del mismo conjunto de puntos, ya que el número sólo depende de  $n$  y  $h$



# Maximización de ángulos e intercambio de aristas

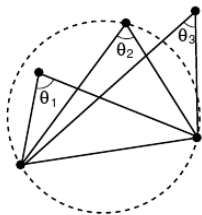
- **Teorema:** Entre todas las posibles triangulaciones de un conjunto planar de puntos dado, la triangulación de Delaunay tiene la secuencia lexicográfica de ángulos más grande, y en particular, ésta maximiza el ángulo mínimo
- Antes de comenzar la demostración de este teorema, debemos recordar algunos hechos básicos de geometría elemental acerca de los ángulos
- En primer lugar, recordemos que si consideramos la circunferencia circunscrita de tres puntos, cada ángulo del triángulo resultante es exactamente la mitad del ángulo del arco menor formado por los dos puntos opuestos





# Maximización de ángulos e intercambio de aristas

- De ello se deduce también que si un punto está dentro de este círculo entonces formará un mayor ángulo y un punto que está fuera formará un ángulo más pequeño
- En la siguiente figura tenemos que  $\theta_1 > \theta_2 > \theta_3$



# Maximización de ángulos e intercambio de aristas

- No vamos a dar una prueba formal del teorema (ver libro de texto), en vez de eso explicaremos la idea principal de ésta
- La idea principal es mostrar que para cualquier triangulación que no cumple la propiedad del círculo vacío, es posible realizar una operación local, denominada *intercambio de aristas*, la cual aumenta la secuencia lexicográfica de los ángulos
- El intercambio de aristas es una operación fundamental de las triangulaciones en el plano



# Maximización de ángulos e intercambio de aristas

- Dados dos triángulos adyacentes  $\triangle abc$  y  $\triangle cda$ , tales que su unión forma un cuadrilátero convexo  $abcd$ , la operación de intercambio de aristas reemplaza la diagonal  $ac$  con  $bd$
- Tenga en cuenta que sólo es posible cuando el cuadrilátero es convexo
- Supongamos que el par de triángulos inicial viola la condición del círculo vacío, i.e. el punto  $d$  se encuentra en el interior de la circunferencia circunscrita de  $\triangle abc$
- Tenga en cuenta que esto implica que  $b$  se encuentra en el interior de la circunferencia circunscrita de  $\triangle cda$



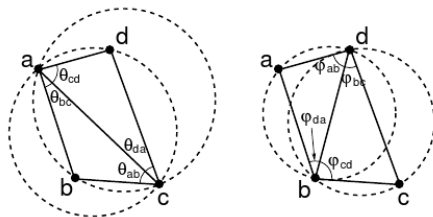
# Maximización de ángulos e intercambio de aristas

- Si intercambiamos la arista las dos circunferencias circunscritas de los dos triángulos resultantes,  $\triangle abd$  y  $\triangle bcd$  están vacías (en relación a estos cuatro puntos)
- Y la observación anterior acerca de los círculos y ángulos demuestra que el ángulo mínimo aumenta al mismo tiempo



# Maximización de ángulos e intercambio de aristas

- En particular, en la siguiente figura, tenemos que  $\varphi_{ab} > \theta_{ab}$ ,  
 $\varphi_{bc} > \theta_{bc}$ ,  $\varphi_{cd} > \theta_{cd}$ ,  $\varphi_{da} > \theta_{da}$



# Maximización de ángulos e intercambio de aristas

- No es difícil mostrar que, tras el intercambio de aristas, estos otros dos ángulos no puede ser menores que el mínimo de  $\theta_{ab}$ ,  $\theta_{bc}$ ,  $\theta_{cd}$ , y  $\theta_{da}$
- Dado que sólo hay un número finito de triangulaciones, este proceso debe terminar finalmente con la triangulación léxicográfica máxima
- Esta triangulación debe cumplir la condición del círculo vacío, y por lo tanto será la triangulación de Delaunay
- Tenga en cuenta que el proceso de intercambio de aristas pueden generalizarse a simplíces complejos en dimensiones mayores



# Maximización de ángulos e intercambio de aristas

- Sin embargo, el proceso no suele sustituir un número fijo de triángulos con el mismo número, como se hace en el plano (dos triángulos por dos nuevos triángulos)
- Por ejemplo, en un 3-espacio, el intercambio de aristas más básico puede reemplazar dos tetraedros adyacentes con tres tetraedros, y viceversa
- Aunque se sabe que en el plano cualquier triangulación pueden convertirse en cualquier otra a través de una secuencia de intercambio de aristas, esto no ha sido comprobado en dimensiones mayores



- 1 Triangulaciones de Delaunay
- 2 Algoritmo de construcción incremental
  - Introducción
  - Prueba en círculo
  - Actualización incremental
  - Localización de un punto
  - Cambios estructurales
  - Reagrupamiento de sitios





# Introducción

- Ahora presentaremos un algoritmo aleatorizado simple para construir triangulaciones de Delaunay para  $n$  sitios en el plano, el cual tiene un tiempo esperado de ejecución  $O(n \log n)$
- El algoritmo es muy similar al algoritmo aleatorizado para mapas trapezoidales en el sentido en que no sólo construye la triangulación, sino que también proporciona una estructura de datos para localización de puntos
- No vamos a discutir a fondo esta estructura de datos, sin embargo los detalles son fáciles de deducir



# Introducción

- Como ocurre con cualquier algoritmo aleatorio incremental, la idea es insertar sitios en orden aleatorio, uno a la vez, y actualizar la triangulación con cada nueva adición
- Las cuestiones relacionadas con el análisis mostrarán que después de cada inserción el número esperado de cambios estructurales en el diagrama es  $O(1)$
- Al igual que con otros algoritmos incrementales, necesitamos alguna forma de llevar un control de dónde han sido agregados los nuevos sitios en el diagrama



# Introducción

- Vamos a describir un método un poco más sencillo que el que usamos para los mapas trapezoidales
- En lugar de construir una estructura de datos, este método simplemente pone cada uno de los puntos aún no insertados en un grupo (racimo) de acuerdo con el triángulo que lo contiene en la triangulación actual
- En este caso, podemos afirmar que el número de veces que un sitio es reagrupado es  $O(\log n)$



- 1 Triangulaciones de Delaunay
- 2 Algoritmo de construcción incremental
  - Introducción
  - Prueba en círculo
  - Actualización incremental
  - Localización de un punto
  - Cambios estructurales
  - Reagrupamiento de sitios



# Prueba en círculo

- La cuestión principal en el diseño del algoritmo es la forma de actualizar la triangulación cuando un nuevo sitio es agregado
- Para ello debemos recordar algunas propiedades básicas de una triangulación de Delaunay
- Recordemos que un triángulo  $\triangle abc$  está en la triangulación de Delaunay, si y sólo si la circunferencia circunscrita de este triángulo no contiene ningún otro sitio en su interior



# Prueba en círculo

- ¿Cómo comprobar si un sitio  $d$  cae en el interior de la circunferencia circunscrita de  $\triangle abc$ ?
- Resulta que este problema puede ser reducido al cálculo de un determinante
- En primer lugar, vamos a suponer que la secuencia  $\langle abcd \rangle$  define un polígono convexo



# Prueba en círculo

- Si no lo hace porque  $d$  se encuentra dentro del triángulo  $\triangle abc$ , entonces  $d$  claramente cae en la circunferencia circunscrita de este triángulo
- Bajo esta suposición,  $d$  cae en la circunferencia circunscrita determinada por el triángulo  $\triangle abc$  si y sólo si el siguiente determinante es positivo

$$\text{enCírculo}(a, b, c, d) = \det \begin{pmatrix} a_x & a_y & a_x^2 + a_y^2 & 1 \\ b_x & b_y & b_x^2 + b_y^2 & 1 \\ c_x & c_y & c_x^2 + c_y^2 & 1 \\ d_x & d_y & d_x^2 + d_y^2 & 1 \end{pmatrix}$$



- 1 Triangulaciones de Delaunay
- 2 **Algoritmo de construcción incremental**
  - Introducción
  - Prueba en círculo
  - **Actualización incremental**
  - Localización de un punto
  - Cambios estructurales
  - Reagrupamiento de sitios





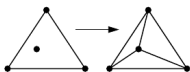
# Actualización incremental

- Cuando agregamos el siguiente sitio,  $p_i$ , el problema consiste en convertir la triangulación actual en una nueva triangulación de Delaunay que contenga este sitio
- Comúnmente al agregar el nuevo sitio  $p_i$  la triangulación actual deja de tener las propiedades de una triangulación de Delaunay
- Por esta razón debe ser reparada incrementalmente hasta lograr que esta triangulación cumpla con dichas propiedades



# Actualización incremental

- Los cambios fundamentales para reparar una triangulación son 2:
  1. Agregar un sitio a la mitad de un triángulo, creando tres nuevas aristas
  2. Efectuar un intercambio de aristas



- Ambas operaciones se pueden realizar en un tiempo  $O(1)$ , asumiendo que la triangulación está almacenada en una estructura de datos adecuada (lista de aristas doblemente conectada)

# Actualización incremental

- El algoritmo que vamos a describir fue propuesto en 1992 por Guibas, Knuth, y Sharir en su artículo “*Randomized incremental construction of Delaunay and Voronoi diagrams*”
- El algoritmo se inicia en una primera triangulación tal que todos los puntos se encuentren en la cubierta convexa
- Esto se puede lograr encerrando los puntos en un triángulo lo suficientemente grande

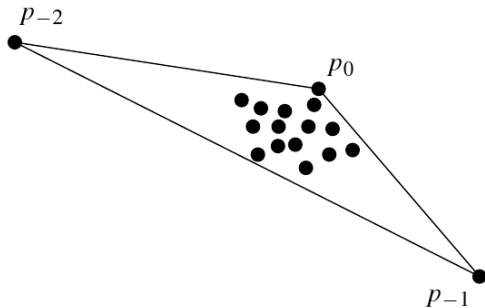


# Actualización incremental

- Debe tenerse cuidado en la construcción de este triángulo, pues no es suficiente que simplemente contenga todos los puntos
- Debe ser tan grande que los vértices del triángulo no caigan en la circunferencia circunscrita de ninguno de los triángulos de la triangulación final
- Nuestro libro de texto sugiere una alternativa, la cual consiste en calcular un triángulo que contiene todos los puntos, pero después realizar un truco a la operación *enCírculo* de modo que los vértices de este triángulo se comporten como si estuvieran infinitamente lejos

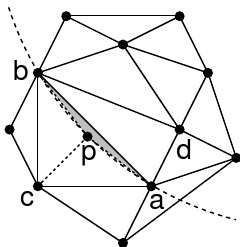


# Actualización incremental



# Actualización incremental

- En este algoritmo los sitios se agregan en orden aleatorio
- Cuando un nuevo sitio  $p$  se agrega, se busca el triángulo  $\triangle abc$  de la triangulación actual que contiene este sitio (veremos cómo hacer esto más adelante), se añade el sitio en este triángulo, y se une este sitio a los tres vértices que lo rodean
- Esto crea tres nuevos triángulos,  $\triangle pab$ ,  $\triangle pbc$ , y  $\triangle pca$ , cada uno de los cuales pueden o no satisfacer la condición del círculo vacío



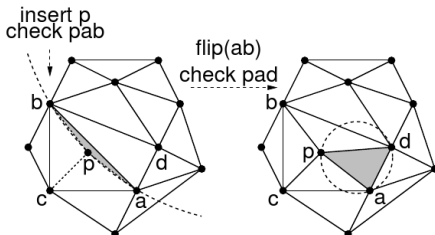
# Actualización incremental

- ¿Cómo podemos verificar esto?
- Para cada uno de los triángulos que se han agregado, verificamos el vértice del triángulo que se encuentra en el lado opuesto de la arista que no incluye  $p$
- Si no existe tal vértice, porque esta arista se encuentra en la cubierta convexa, entonces el proceso termina



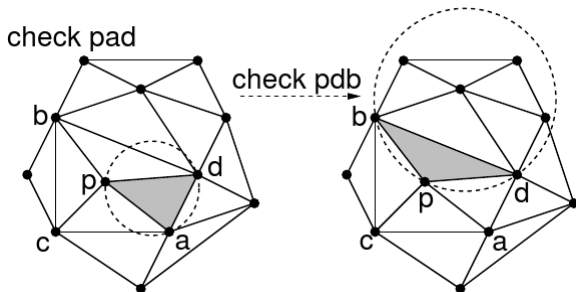
# Actualización incremental

- Si este vértice falla la prueba *enCírculo* (i.e., si está dentro de la circunferencia circunscrita), entonces la arista se intercambia reemplazando un triángulo que era adyacente a  $p$  con dos nuevos triángulos
- Este proceso se repite con estos dos nuevos triángulos

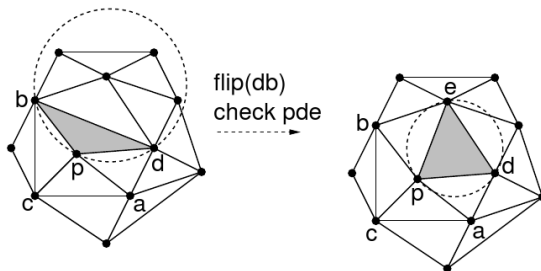




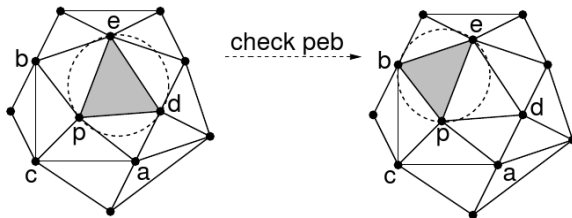
# Actualización incremental



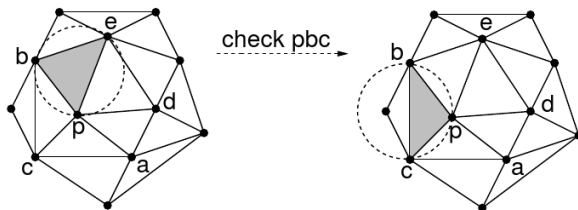
# Actualización incremental



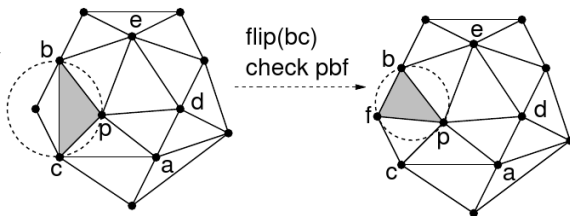
# Actualización incremental



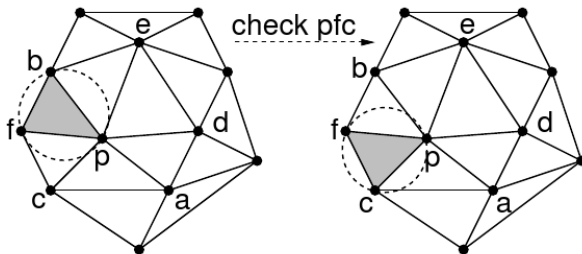
# Actualización incremental



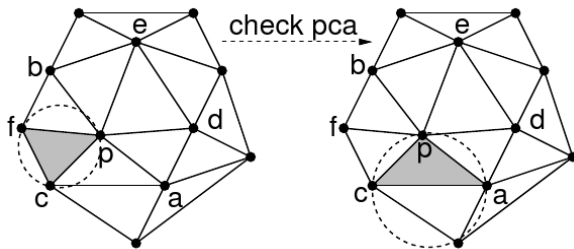
# Actualización incremental



# Actualización incremental



# Actualización incremental



# Actualización incremental

\_\_\_\_\_ Randomized Incremental Delaunay Triangulation Algorithm

```

Insert( $p$ ) {
    Find the triangle  $\triangle abc$  containing  $p$ ;
    Insert edges  $pa$ ,  $pb$ , and  $pc$  into triangulation;
    SwapTest( $ab$ );           // Fix the surrounding edges
    SwapTest( $bc$ );
    SwapTest( $ca$ );
}

SwapTest( $ab$ ) {
    if ( $ab$  is an edge on the exterior face) return;
    Let  $d$  be the vertex to the right of edge  $ab$ ;
    if (inCircle( $p, a, b, d$ ) {           //  $d$  violates the incircle test
        Flip edge  $ab$  for  $pd$ ;
        SwapTest( $ad$ );           // Fix the new suspect edges
        SwapTest( $db$ );
    }
}

```





# Actualización incremental

- Como se puede ver, el algoritmo es muy simple
- Sólo hay dos elementos de este algoritmo que no se han descrito hasta el momento
- El primero son las operaciones de actualización sobre la estructura de datos que representa el símplex complejo
- Estas se puede realizar en tiempo  $O(1)$  cada una usando una representación adecuada
- El otro elemento es la localización del triángulo que contiene  $p$ , esto lo explicaremos a continuación



- 1 Triangulaciones de Delaunay
- 2 **Algoritmo de construcción incremental**
  - Introducción
  - Prueba en círculo
  - Actualización incremental
  - **Localización de un punto**
  - Cambios estructurales
  - Reagrupamiento de sitios



# Localización de un punto

- La ubicación de un punto puede ser efectuada de dos formas
- Nuestro libro de texto analiza la idea de construir una estructura de datos para localización de puntos basada en un grafo, tal como lo hicimos en el caso de los mapas trapezoidales
- Un enfoque más sencillo se basa en la idea de mantener los sitios aún no insertados en un conjunto de grupos (racimos)



## Localización de un punto

- Imaginemos cada triángulo de la triangulación actual como un grupo que contiene los sitios que se encuentran dentro de este triángulo y aún no se han insertado
- Siempre que una arista sea intercambiada, o cuando un triángulo sea dividido en tres triángulos al insertar un punto, algunos antiguos triángulos son destruidos y sustituidos por un número constante de nuevos triángulos
- Cuando esto ocurre, reunimos todos los sitios en los grupos correspondientes a los triángulos suprimidos, creamos nuevos grupos para los triángulos recién creados, y reasignamos cada sitio en su nuevo grupo



# Localización de un punto

- Puesto que hay un número constante de triángulos creado, este proceso requiere un tiempo  $O(1)$  por cada sitio que se reagrupa
- Para analizar el tiempo esperado de ejecución del algoritmo necesitamos acotar dos cantidades:
  - 1 ¿cuántos cambios estructurales son realizados en la triangulación en promedio con la adición de cada nuevo sitio
  - 2 ¿cuánto esfuerzo es necesario para reagrupar los sitios?
- Como de costumbre, nuestro análisis será para el peor de los casos (para cualquier conjunto de puntos), pero promediado sobre todos los posibles órdenes de inserción



- 1 Triangulaciones de Delaunay
- 2 **Algoritmo de construcción incremental**
  - Introducción
  - Prueba en círculo
  - Actualización incremental
  - Localización de un punto
  - **Cambios estructurales**
  - Reagrupamiento de sitios



# Cambios estructurales

- Como habíamos mencionado el número esperado de aristas que cambian con cada inserción es  $O(1)$ , esto se puede deducir utilizando un simple análisis inverso
- En primer lugar observemos que (suponiendo la posición general) la estructura de la triangulación de Delaunay es independiente del orden de los sitios insertados hasta el momento
- Por lo tanto, cualquiera de los sitios existentes es igualmente probable que haya sido el último en ser agregado a la estructura
- Supongamos que un sitio  $p$  es el último que se ha insertado, ¿Cuánto trabajo se requirió para insertar  $p$ ?



# Cambios estructurales

- Observe que la inserción inicial de  $p$  implicaría la creación de tres nuevas aristas, todas incidentes a  $p$
- Además, siempre que una arista es intercambiada, una nueva arista es agregada a  $p$
- Estos son los únicos cambios que el algoritmo de inserción puede hacer
- Por lo tanto, el número total de cambios realizados en la triangulación para la inserción de  $p$  es proporcional al grado de  $p$  *después* de que la inserción se ha completado





# Cambios estructurales

- Por ello, el trabajo necesario para insertar  $p$  es también proporcional al grado de  $p$  después de la inserción
- Así el tiempo esperado para insertar el  $i$ -ésimo sitio es igual al grado promedio de un vértice en la triangulación de  $i$  sitios
- La única excepción son los tres primeros vértices en el infinito, que deben ser los primeros sitios que se insertan
- Mediante la fórmula de Euler, sabemos que el grado promedio de un vértice en un grafo planar es como máximo 6



# Cambios estructurales

- Esto porque un grafo planar con  $n$  vértices pueden tener como máximo  $3n$  aristas, y la suma de los grados de los vértices es igual a dos veces el número de aristas, el cual es como máximo  $6n$
- Así pues, independientemente del número de iteración del algoritmo, el número esperado de cambios de aristas es proporcional al grado esperado de los vértices, que es  $O(1)$
- Sumando todas las  $n$  inserciones, el número total de cambios estructurales es  $O(n)$
- Recordemos que cada uno de los cambios estructurales (nuevas aristas e intercambio de aristas) se pueden realizar en tiempo  $O(1)$



- 1 Triangulaciones de Delaunay
- 2 Algoritmo de construcción incremental
  - Introducción
  - Prueba en círculo
  - Actualización incremental
  - Localización de un punto
  - Cambios estructurales
  - Reagrupamiento de sitios



# Reagrupamiento de sitios

- A continuación mostraremos que el tiempo total esperado que se utiliza en operaciones de *reagrupamiento de sitios* es  $O(n \log n)$
- Para ello, mostraremos que el número esperado de veces que cualquier sitio es reagrupado es  $O(\log n)$
- Una vez más esto se hace aplicando un análisis inverso estándar
- Vamos a fijar un sitio  $q \in P$



# Reagrupamiento de sitios

- Consideremos la situación justo después de la inserción del  $i$ -ésimo sitio
- Si  $q$  ya se ha insertado, entonces no está implicado en el proceso de reagrupamiento, así que asumamos que  $q$  aún no ha sido insertado
- Como en el caso anterior hacemos uso del hecho de que cualquiera de los sitios es igualmente probable que haya sido el último en ser agregado
- Comencemos calculando la probabilidad de que  $q$  sea reagrupado como resultado de la última inserción



# Reagrupamiento de sitios

- Para ello, designemos como  $\tau$  el triángulo que contiene  $q$  después de la  $i$ -ésima inserción
- Como se ha señalado anteriormente, después de insertar el  $i$ -ésimo sitio todos los triángulos recién creados son incidentes a este nuevo sitio
- $\tau$  se origina como resultado de la última inserción si y sólo si uno de sus tres vértices fue el último en ser insertado
- Dado que  $\tau$  es incidente a exactamente tres sitios, y cada sitio es igualmente probable que sea el último insertado, se deduce entonces que la probabilidad de que  $\tau$  exista es igual a  $3/i$



# Reagrupamiento de sitios

- Así, la probabilidad de que  $q$  requiera ser reagrupado después de la última inserción es como máximo igual a  $3/i$
- Después de la  $i$ -ésima inserción hay  $n - i$  puntos que podrían ser objeto de un reagrupamiento, y cada uno tiene probabilidad  $3/i$
- Por lo que el número esperado de puntos que requieren reagrupamiento como parte de la última inserción es  $(n - i)3/i$



# Reagrupamiento de sitios

- Dada la linealidad del valor esperado, para obtener el número total de reagrupamientos, se suman sobre todas las inserciones dando como resultado:

$$\sum_{i=1}^n \frac{3}{i}(n-i) \leq \sum_{i=1}^n \frac{3}{i}n = 3n \sum_{i=1}^n \frac{1}{i} = 3n \ln n + O(1)$$

- Por lo que el tiempo total esperado que se utiliza en reagrupamiento de sitios es  $O(n \log n)$
- De este hecho es fácil observar que el tiempo de total consumido por el algoritmo para construir una triangulación de Delaunay está dominado por el proceso de reagrupamiento de sitios, por lo que éste también es  $O(n \log n)$

