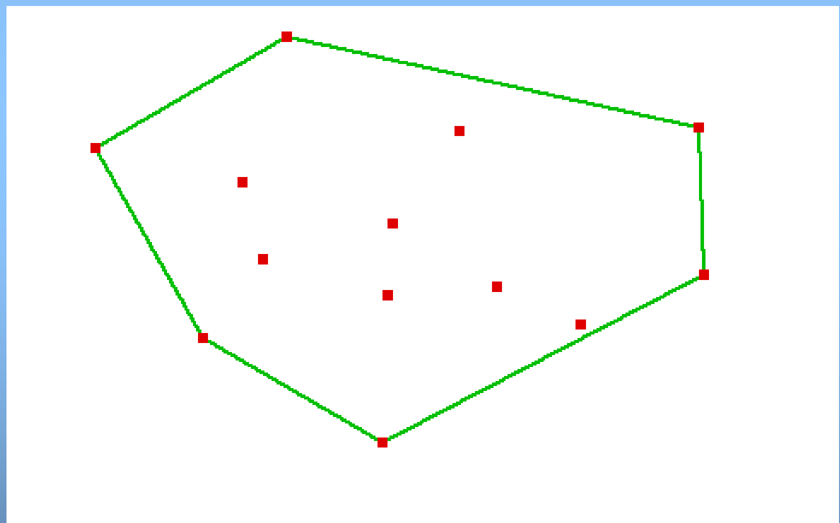
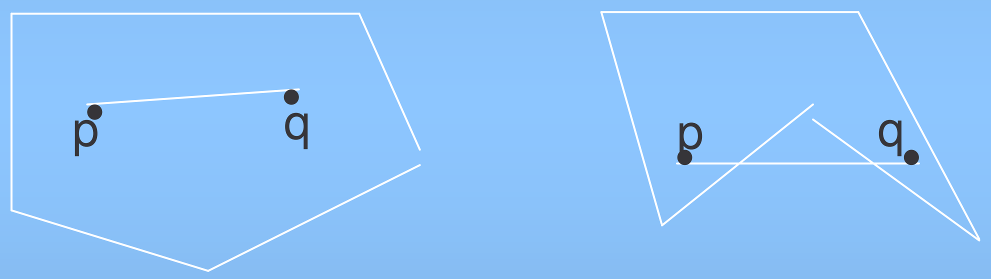
Envolventes convexas



Envolventes Convexas I

Definición: Un conjunto C del plano afín es convexo si dados dos puntos cualesquiera p y q de C, el segmento que une p y q está contenido en C.

Segmento: $[p,q]=(1-t)p+tq,\ t\in[0,1]$



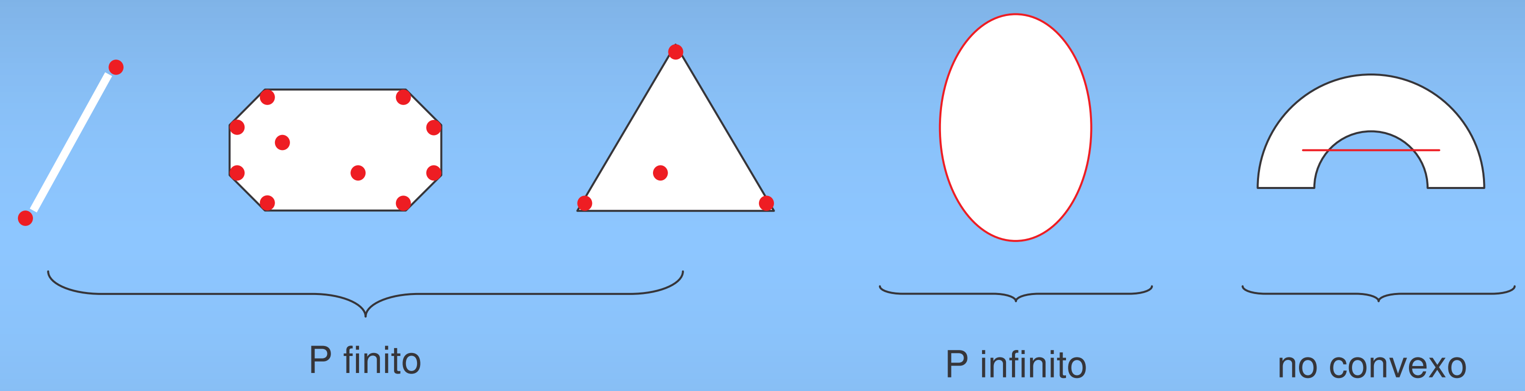
convexo **no convexo**

Nota: la intersección de conjuntos convexos es convexo

Definición: La envolvente convexa de un conjunto de puntos P es el menor conjunto convexo que contiene a dichos puntos (es decir, la intersección de todos los conjuntos convexos en los que P está contenido).

Envolventes Convexas II

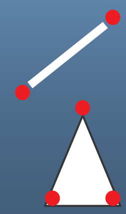
Definición: La envolvente convexa de un conjunto de puntos P es el menor conjunto convexo que contiene a dichos puntos.



Resultado: Dado $P=\{p_1, \ldots , p_n\}$, $C(P)$ coincide con el conjunto de las combinaciones convexas de los elementos de P, es decir,

$$\boxed{C(P)=\{a_1p_1+\ldots+a_np_n \setminus a_1+\ldots+a_n=1,\ a_i \geq 0\}}$$

Casos particulares:
- C formado por dos puntos: C(P) es un segmento
- C formado por tres puntos: C(P) es un triángulo

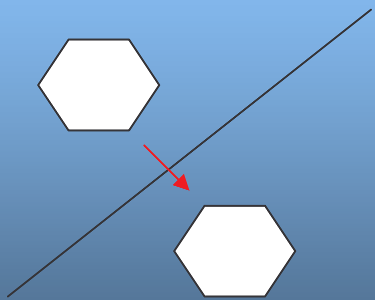


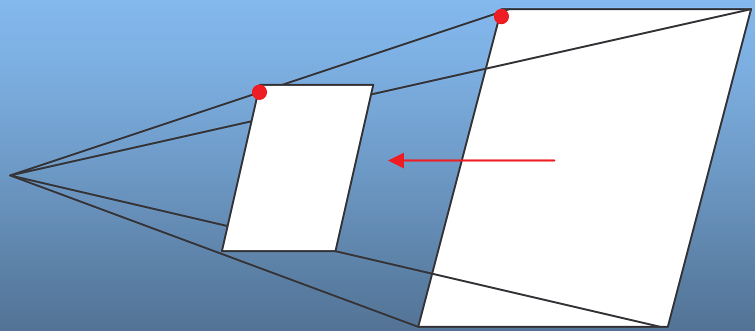
Envolventes Convexas III

Resultado: **Como las transformaciones afines respetan combinaciones convexas, si f es una transformación afín,**

$$\boxed{f(C(\{p_1,\ldots, p_n\}))=C(\{f(p_1), \ldots, f(p_n)\})}$$

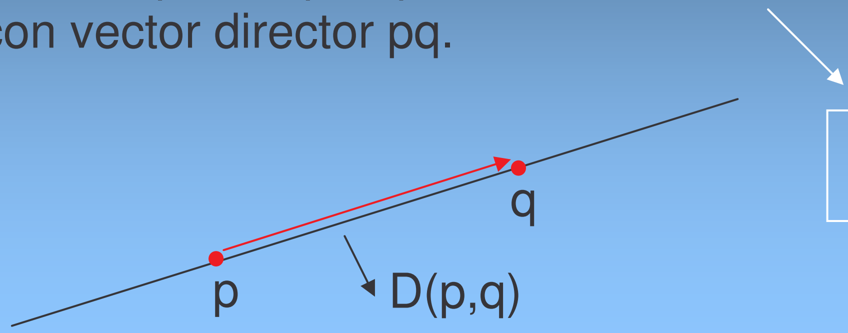
Conclusión: basta girar, trasladar, hacer simetrías, homotecias, etc., con los puntos de P y luego hallar su envolvente convexa





Envolventes Convexas IV

Llamamos **D(p,q)** al semiplano que queda a la derecha de la recta que determinan los puntos p y q, con vector director pq.

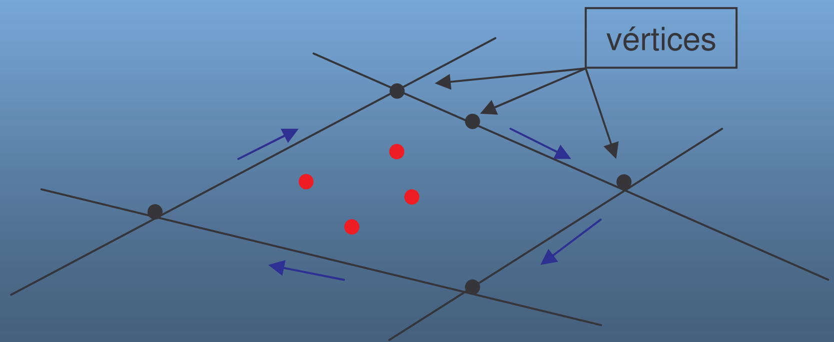


se comprueba estudiando el signo de un determinante

Resultado: Si $P=\{p_1, \dots, p_n\}$ y no todos ellos están alineados, entonces

$$\boxed{C(P)=D(q_1,q_2)\cap D(q_2,q_3)\cap \dots \cap D(q_{n-1},q_n),}$$

donde $q_1, \dots, q_n$ son elementos de P y se llaman vértices de la envolvente.



Todos los puntos forman P, pero los que están en negro, son vértices

Envolventes Convexas V

Primer algoritmo:

Entrada: P
Salida: Q (vértices ordenados)

Paso 1: tomar todas las parejas de puntos de P ← *complejidad* O(n^2)
Paso 2: para cada para cada pareja (p_i,p_j), ← *complejidad* O(n^2)× O(n)
- si todos los demás puntos de P están en D(p_i,p_j) y
- no hay más puntos de P en el segmento [p_i,p_j]

SE ALMACENA (p_i, p_j)
- en caso contrario, se pasa a la pareja siguiente.

Paso 3: extraer ordenadamente los vértices ← *complejidad* O(n log (n))
- si están todos alineados, escoger el de un extremo (orden lexicográfico) y partir de él para hallar los demás.
- en caso contrario, empezar por uno cualquiera y hacer (q_1,q_2)(q_2, q_3)(q_3,q_4)… Extraer q_1,q_2,q_3,q_4….

Complejidad (en el peor de los casos): **O(n^3)**

Envolventes Convexas VI

Algoritmo basado en giros a derecha:

Entrada: P
Salida: Q (vértices ordenados)

Paso 1: Semienvolvente superior (dos listas: auxiliar, final)

1. Ordenar los puntos en orden lexicográfico ← *complejidad* O(n log(n))
   Inicializar auxiliar:=$p_1,\ldots, p_n$
2. Tomar los tres primeros puntos:
   final:=p_1,p_2,p_3
   auxiliar:=$p_4,\ldots,p_n$
3. Comprobar si los tres últimos elementos de final hacen un giro a derecha: ←O(n)
   SI: añadir a final el siguiente elemento de auxiliar
   NO: eliminar de final su penúltimo elemento.
4. Repetir el proceso hasta que final tenga la semienvolvente superior.

Paso 2: Semienvolvente inferior: análogo, cambiando el orden

Paso 3: Combinar los resultados de los pasos anteriores y extraer los vértices.

Complejidad (en el peor de los casos): **O(n log(n))**

Envolventes Convexas VII

Lecturas adicionales:

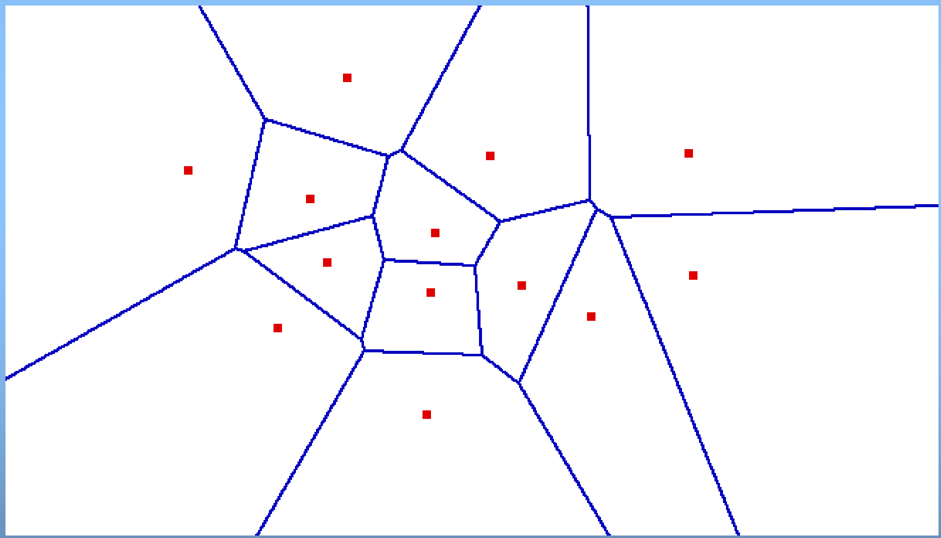
Otros algoritmos: (en 2 dimensiones)

1. Gift wrapping (Chand & Kapur, 1970)
2. Quickhull (Preparata & Shamos, 1985)
3. Algoritmo de Graham (Graham, 1972)
4. Algoritmo Incremental
5. Divide y vencerás (Preparata & Hong, 1977)

Envolventes convexas en 3 dimensiones

Bibliografía recomendada: “Computational Geometry in C”, Joseph O’Rourke
http://www.cse.unsw.edu.au/~lambert/java/3d/hull.html

Diagramas de Voronoi



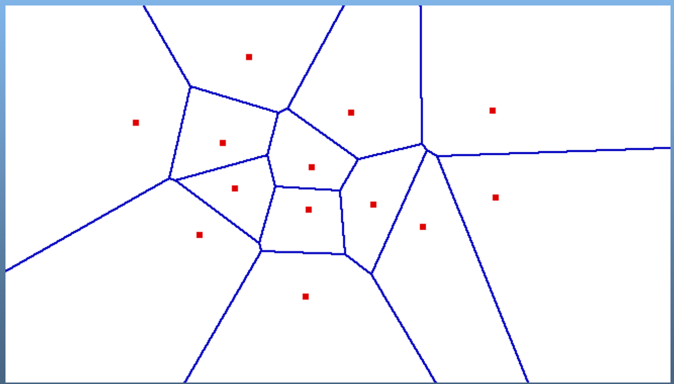
Diagramas de Voronoi I

Definición: Dada una familia finita de puntos $P=\{p_1,\ldots,p_n\}$, la celda de Voronoi de p_i es:

$$\boxed{V(p_i)=\{p \;/\; d(p,p_i)<d(p,p_k),\ \forall\, k\neq i\}}$$

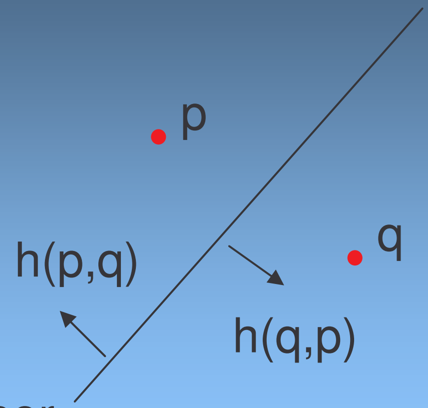
es decir, está formada por los puntos del plano que están más cerca de p_i que del resto de los p_k. En ocasiones, también se llama "zona de influencia" de p_i.

Definición: El diagrama de Voronoi de $P=\{p_1,\ldots,p_n\}$ está formado por las celdas de Voronoi de cada uno de los puntos de P. Se denota Vor(P).

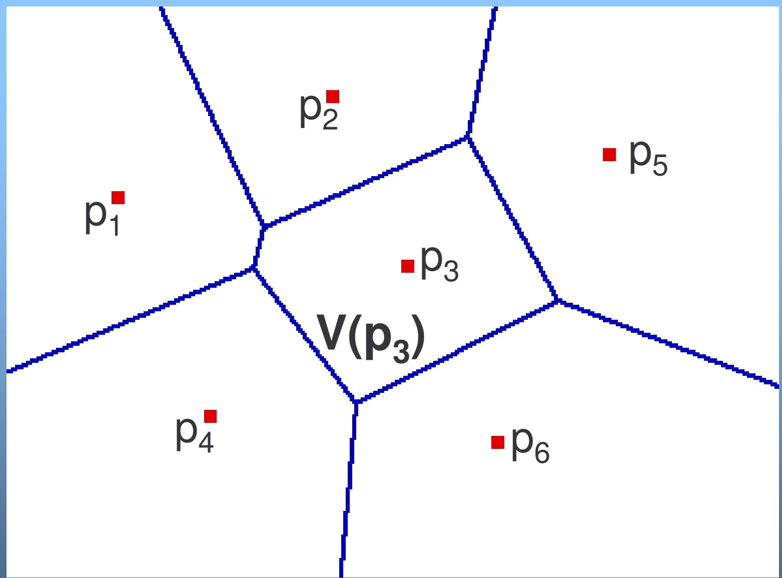


Diagramas de Voronoi II

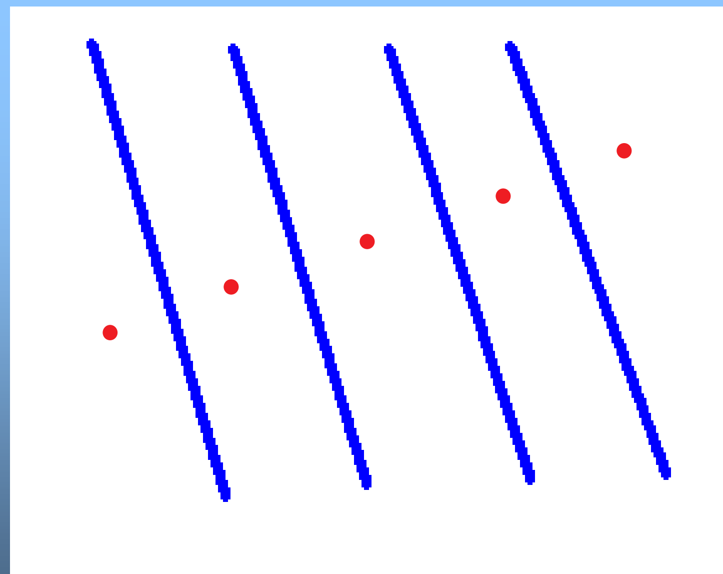
Si P={p,q}, definimos h(p,q)={x / d(x, p)< d(x,q)}
h(q,p)={x / d(x, q)< d(x,p)}



Resultado: Cada celda de Voronoi $V(p_i)$ está formada por la intersección de todos los $h(p_i, p_k)$.



caso no degenerado



caso degenerado
(todos los puntos alineados)

Diagramas de Voronoi III
(caso no degenerado)

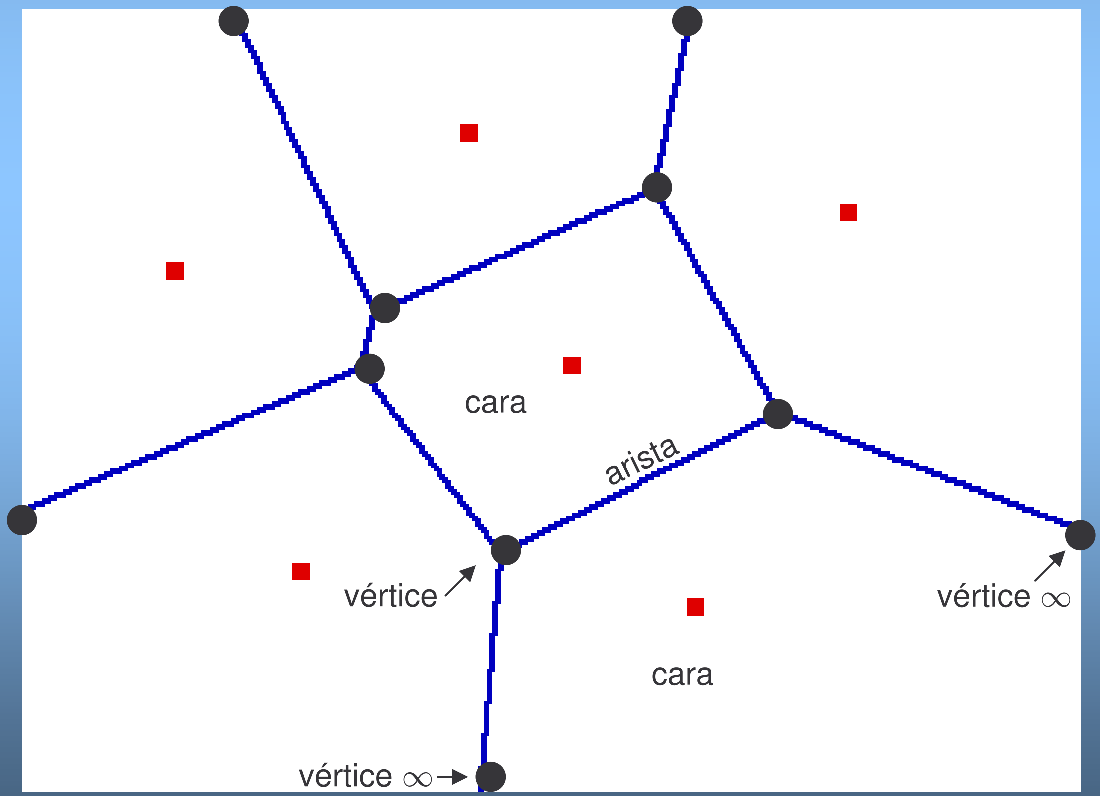
Definición: cada celda de Voronoi es un conjunto convexo, limitado por:

- **vértices de Voronoi: puede ser ∞**
- **aristas de Voronoi: unen vértices**

→ forman un grafo plano conexo

↓

- **cada $V(p_i)$ es una cara del grafo**
- **Fórmula de Euler: $|V|+|C|-|A|=2$**



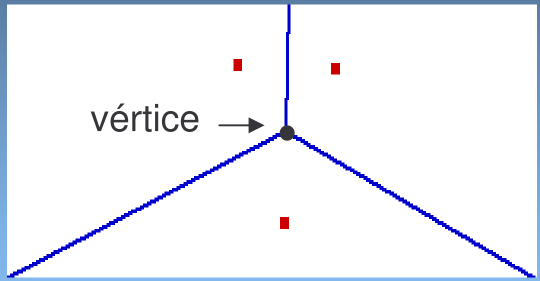
- **aristas:** corresponden a 2 puntos
- **vértices:** corresponden a 3 puntos

se calcula el grafo asociado a Vor(P)

Diagramas de Voronoi IV

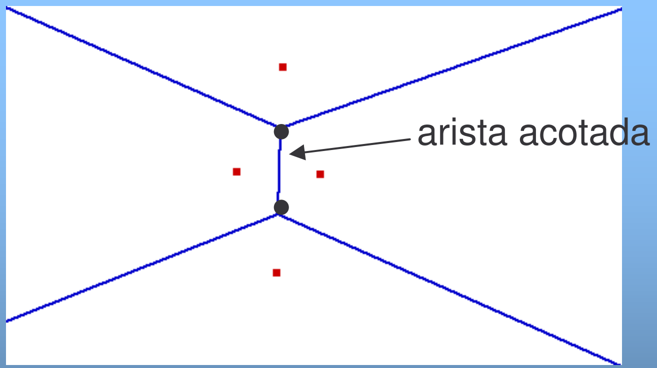
(caracterización de vértices y aristas)

Caracterización de vértices: son el centro de una circunferencia que pasa por tres puntos y deja los demás puntos fuera.



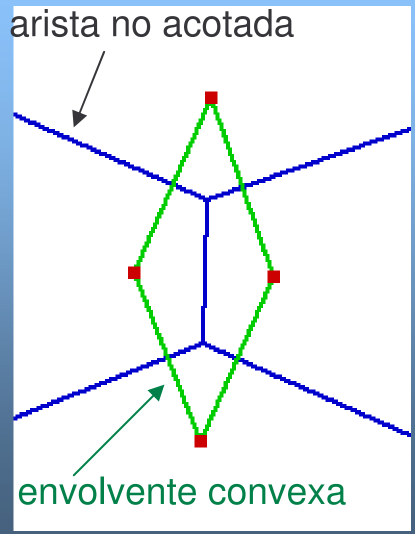
$|V| \leq 2n-5$

Caracterización de aristas acotadas: están entre dos vértices y desde su punto medio se puede trazar una circunferencia que pasa por dos puntos y deja los demás fuera.



$|A| \leq 3n-6$

Caracterización de aristas no acotadas: son meditrices entre puntos consecutivos de la envolvente convexa. Parten de un vértice y se orientan hacia fuera.



Diagramas de Voronoi V

Algoritmo basado las caracterizaciones de vértices y aristas:

Entrada: P
Salida: (V,A) (grafo plano asociado a Vor(P))

Paso 1: Vértices

- Hacemos todas las ternas de puntos distintos de P.
- Para cada (p_i,p_j,p_k) hacemos la circunferencia que pasa por ellos,
  - si no hay puntos de P dentro de la circunferencia, su centro es un vértice,
  - en caso contrario, pasar a la siguiente terna.

Complejidad: $O(n^3)\times O(n)=O(n^4)$

Paso 2: Aristas acotadas

- Hacemos todos los pares de vértices.
- Para cada dos vértices, hallar su punto medio m y las circunferencias que tienen centro en m y pasan por dos puntos de P,
  - si dejan fuera los demás puntos de P, es una arista acotada,
  - en caso contrario, pasar al siguiente par de vértices.

Complejidad: $O(n^2)\times O(n^2)\times O(n)=O(n^5)$ ← **Complejidad** (en el peor de los casos)

Paso 3: Aristas no acotadas

- Calcular C(P)
- Hallar las mediatrices de los vértices de C(P)
- Buscar en {vértices} el origen de la arista no acotada (si hay varios, escoger el más exterior).

Complejidad: $O(n\log(n))+O(n)=O(n\log(n))$

Diagramas de Voronoi VI

Trabajo obligatorio para entregar: estudia uno de los siguientes algoritmos óptimos: historia, funcionamiento, pseudocódigo, complejidad

Algoritmo de Fortune: *complejidad* O(n log(n))
"Computational Geometry, algorithms and applications", M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf. Sección 7.2.

Idea: barrido por una recta vertical r
computar en cada acontecimiento la intersección de r con Vor(P)
acontecimientos: "site events" $\mapsto$ aristas
"circle events" $\mapsto$ vértices

Ver su funcionamiento en:

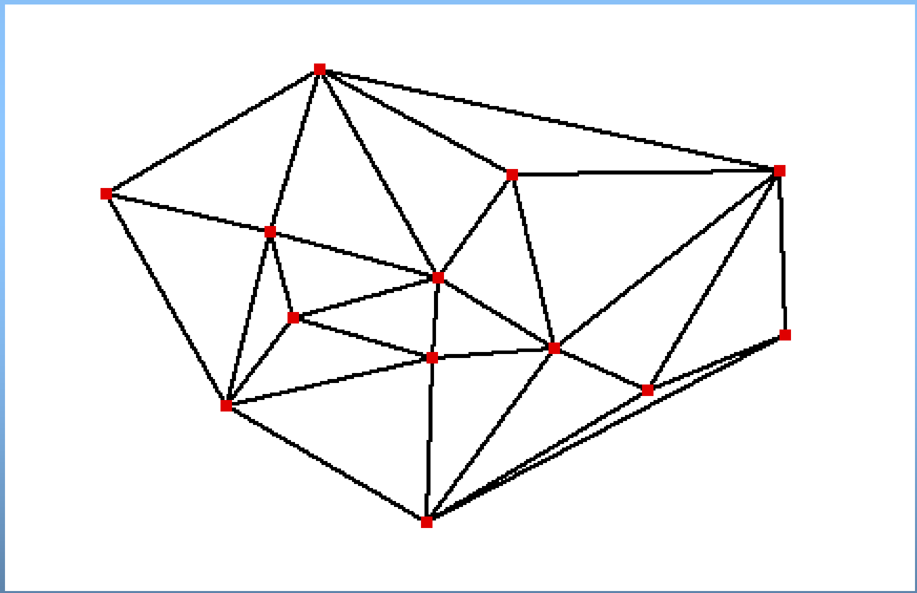
http://www.dma.fi.upm.es/docencia/trabajosfindecarrera/programas/geometriacomputacional/Fortune/voronoi-jar.html

Algoritmo "divide y vencerás": *complejidad* O(n log(n))
"Computational Geometry, an introduction". F. Preparata, M. Shamos. Sección 5.2.2

Páginas web con algoritmos para calcular diagramas de Voronoi:

http://wwwpi6.fernuni-hagen.de/Geometrie-Labor/VoroGlide
http://www.dma.fi.upm.es/docencia/trabajosfindecarrera/programas/geometriacomputacional/

Triangulaciones planas



Triangulaciones

Definición: Dados n puntos del plano $P=\{p_1, \ldots, p_n\}$ no todos alineados, una triangulación T es una subdivisión plana maximal asociada a P, es decir, es un grafo plano (V, A) con

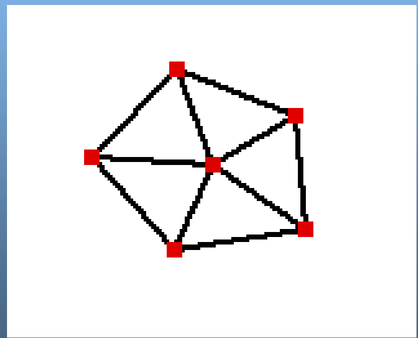
- V=P
- A=segmentos entre vértices de modo que no se puedan añadir más aristas (sin salirse del plano)

Observaciones:
- las caras acotadas son triángulos
- las aristas de C(P) son aristas de A

Resultado: Dado $P=\{p_1, \ldots, p_n\}$ no todos alineados y suponiendo que C(P) tiene k vértices, cada triangulación T de P tiene siempre

número de triángulos: 2n-2-k
número de aristas: 3n-3-k

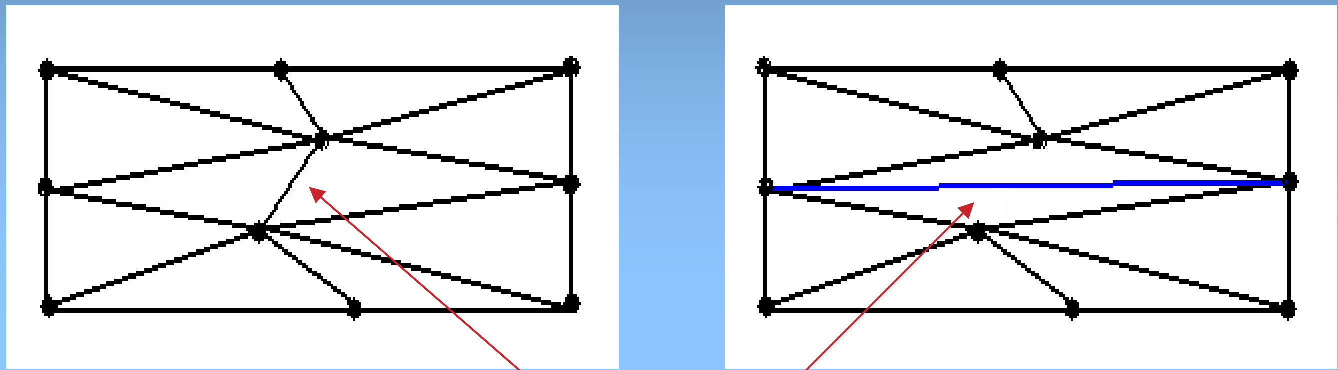
Por la fórmula de Euler



k=5
n=6
$T=2\times n-2-k=5$
$A=3\times n-3-k=10$

Triangulaciones óptimas y legales

Nota: Para modelar la realidad, conviene usar triángulos lo más regulares posibles (lo más equiláteros posibles).



El dibujo izquierdo modeliza mejor la superficie

¿Qué criterio establecemos para comparar triangulaciones? **ÁNGULOS**

Dada una triangulación T, definimos el <u>ángulo de T</u> como la lista ordenada de menor a mayor de los ángulos de todos los triángulos de T

$$\boxed{A(T)=(\alpha_1,\ldots,\alpha_{3m}) \text{ t.q. } \alpha_1\leq \alpha_2\leq\ldots\leq \alpha_{3m}}$$

donde α_i son los ángulos de los triángulos de T

Triangulaciones óptimas

Definición: Dadas dos triangulaciones T y T' de P, decimos que el ángulo de T, A(T)=(α1,…,α_{3m}), es menor que el ángulo de T', A(T')=(α1',…,α_{3m}'), si existe **k**$\leq$ 3m t.q.

$$\boxed{\alpha_1=\alpha_1',\ \alpha_2=\alpha_2',\ \ldots,\ \alpha_{k-1}=\alpha_{k-1}'} \quad \boxed{\alpha_k<\alpha_k'}$$

Ej: (30º,30º,30º,30º,120º,120º) < (60º,60º,60º,60º,60º,60º)

Ej: (30º,30º,30º,90º) < (30º,30º,60º,60º)

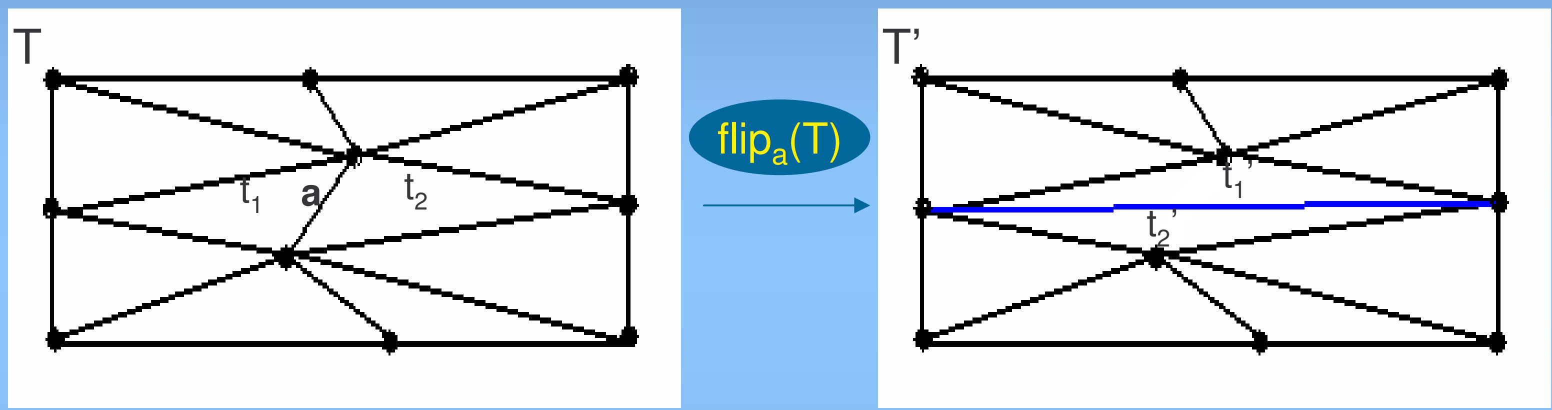
Definición: Una triangulación de P es óptima si su ángulo es mayor o igual que el resto de las triangulaciones de P.

Obs: las triangulaciones óptimas maximizan el ángulo menor y evitan triángulos con ángulos muy agudos.

Triangulaciones legales

Operación flip: Dados dos triángulos t_1 y t_2 que comparten una arista **a** en una triangulación T, la operación $flip_a(T)$ cambia T por otra triangulación T' donde:

- los triángulos fuera del cuadrilátero $t_1 \cup t_2$ son los mismos
- en $t_1 \cup t_2$ tomamos la diagonal opuesta a **a**



Los ángulos de T y de T' son iguales, salvo los 6 ángulos de $t_1 \cup t_2$

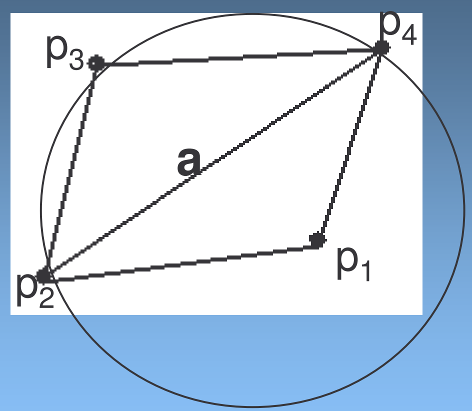
Definición: Una arista **a** es ilegal si al hacer flip_a(T) aumenta el ángulo mínimo $t_1 \cup t_2$.

Definición: Una triangulación es legal si no posee aristas ilegales.

Algoritmo: empezar por una triangulación cualquiera y hacer $flip_a(T)$ por cada una de las aristas ilegales. (el proceso termina en un número finito de pasos)

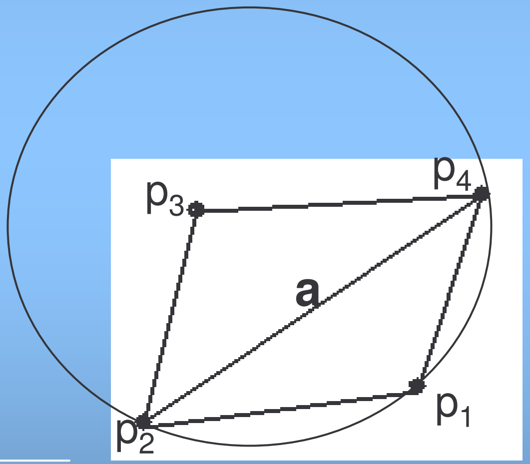
Caracterización de aristas ilegales

Resultado: Una arista es **ilegal** si y sólo si

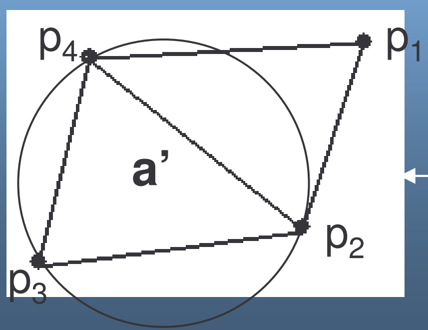


p_1 está en el interior de la circunferencia que pasa por p_2, p_3 y p_4

si y sólo si



p_3 está en el interior de la circunferencia que pasa por p_1, p_2 y p_4

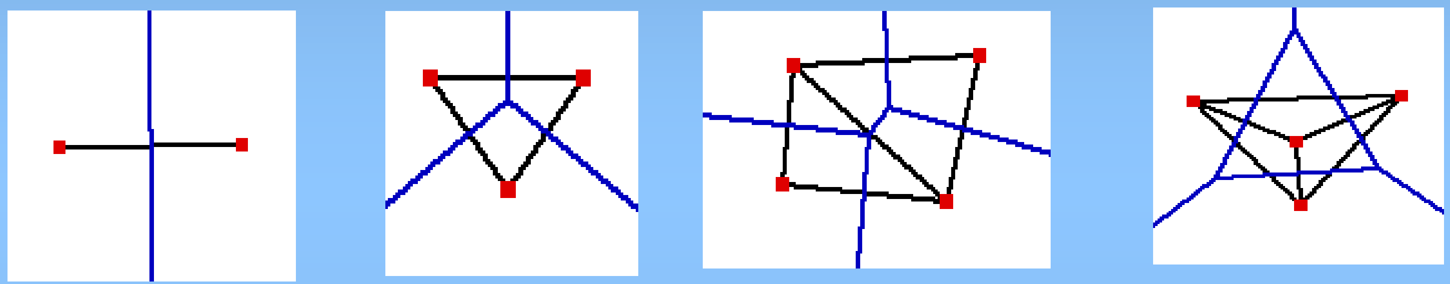


si hacemos $flip_a(T)$ la arista sí es legal

Grafo de Delaunay

Definición: Dado un conjunto $P=\{p_1,\ldots,p_n\}$ de puntos del plano, con diagrama de Voronoi Vor(P), llamamos grafo de Delaunay al grafo dual de Vor(P), es decir,

$$\boxed{DP} \begin{cases} \text{Vértices : } P \\ \text{Aristas: } p_i p_j \text{ siempre que } V(p_i) \text{ y } V(p_j) \text{ compartan una arista} \end{cases}$$



• puntos de P

aristas de Vor(P)

aristas de DP

Caso no degenerado: $n \geq 3$ y no hay cuatro o más puntos formando una circunferencia.

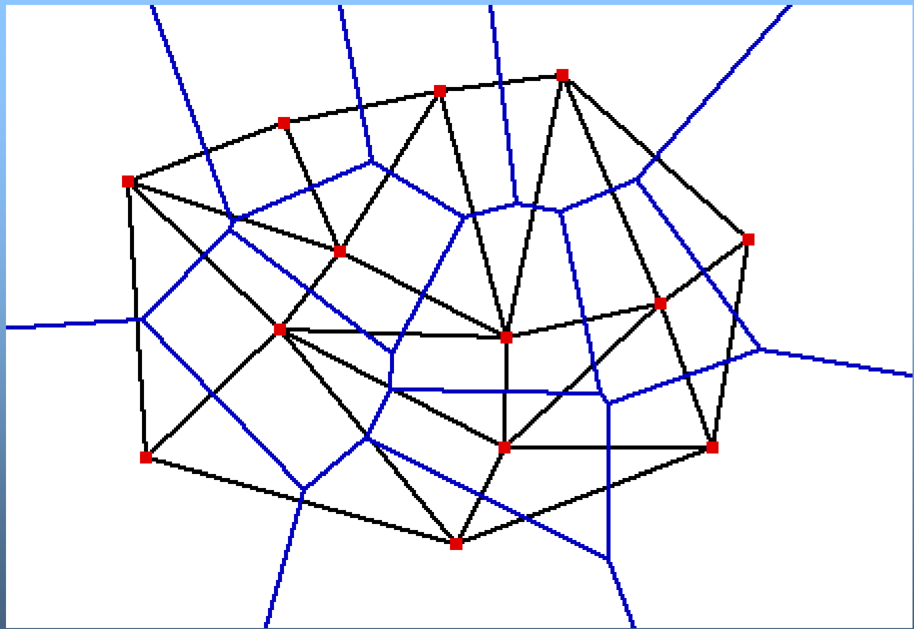
- En el caso no degenerado, DP es una triangulación de P
- En el caso degenerado, DP no es una triangulación, pero podemos añadir aristas para completarlo hasta una triangulación.

Grafo de Delaunay

Resultado 1: En el caso no degenerado, **triangulación óptima** $\Leftrightarrow$ **DP**

Resultado 2: Tanto en el caso degenerado como en el no degenerado, T es una triangulación que se obtiene al completar DP $\Leftrightarrow$ T es **legal**.

Nota: triangulación óptima $\Rightarrow$ triangulación legal



Grafo de Delaunay

Referencias: Este tema está desarrollado con detalle en:

"Computational Geometry, algorithms and applications", M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf. Capítulo 9.

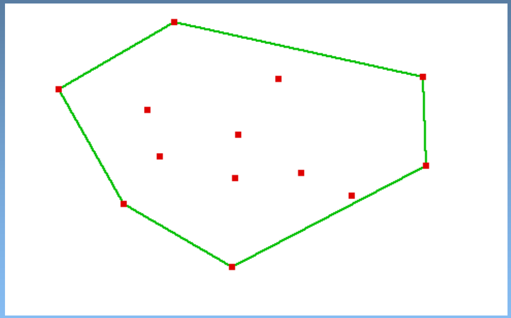
Algoritmos:

1. Usar un algoritmo para hallar Vor(P) y calcular su grafo dual.
2. **Lectura adicional**: en la referencia anterior, secciones 9.3 y 9.4, se describe otro algoritmo para hallar DP y se estudia su complejidad.

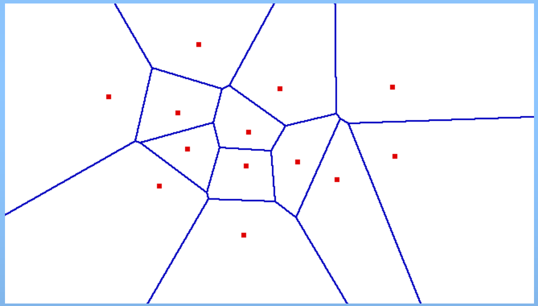
Visita las páginas web:

http://wwwpi6.fernuni-hagen.de/Geometrie-Labor/VoroGlide
http://www.dma.fi.upm.es/docencia/trabajosfindecarrera/programas/geometriacomputacional/

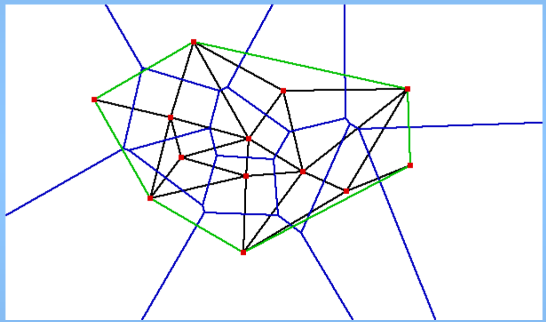
Envolventes convexas, diagramas de Voronoi y triangulaciones de Delaunay

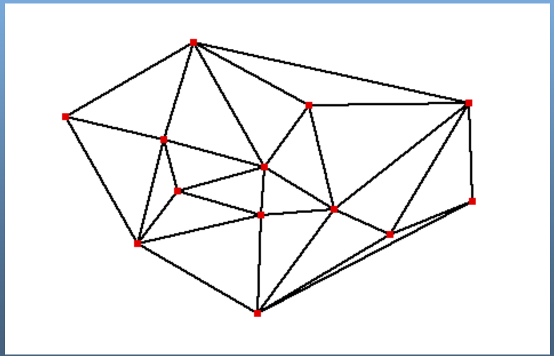


•Envolventes convexas



•Diagramas de Voronoi





•Triangulaciones de Delaunay