Diagramas de Voronoi

Dr. Eduardo A. Rodríguez Tello

Cinvestav-Tamaulipas

15 de marzo del 2013

1. Diagramas de Voronoi
   - Introducción
   - Algunas aplicaciones
   - Propiedades de los diagrama de Voronoi
   - Construcción de diagramas de Voronoi
   - Algoritmo de Fortune
   - Análisis de complejidad

Cinvestav

- El material de esta clase está basado en el capítulo 7 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.

Cinvestav

Cinvestav

Introducción

- Los **diagramas de Voronoi** se encuentran entre las más importantes estructuras en geometría computacional
- Un diagrama de Voronoi codifica la información de proximidad entre elementos
- Sea $P = \{p_1, p_2, \ldots, p_n\}$ un conjunto de puntos en el plano (o en cualquier espacio d-dimensional), los cuales llamaremos **sitios**
- Definimos $\mathcal{V}(p_i)$, la **celda de Voronoi** para p_i, como el conjunto de puntos q en el plano que están más cerca de p_i que de cualquier otro sitio

Cinvestav

Introducción

- Es decir, la celda de Voronoi p_i se define como:

$$\mathcal{V}(p_i) = \{q \mid \|p_i q\| < \|p_j q\|, \forall j \neq i\} \tag{1}$$

- Donde $\|pq\|$ denota la distancia euclídea entre los puntos p y q
- El diagrama de Voronoi puede definirse sobre cualquier métrica y en cualquier dimensión, pero nos concentraremos en el caso planar y Euclidiano

Cinvestav

Introducción

- Otra forma de definir $\mathcal{V}(p_i)$ es en términos de la intersección de semiplanos
- Dados dos sitios p y q en el plano se define la bisectriz de p y q como la mediatriz (perpendicular) del segmento $\overline{pq}$
- Esta bisectriz divide el plano en dos semiplanos
- Denotamos el semiplano abierto que contiene p como $h(p, q)$ y el que contiene q como $h(q, p)$

Cinvestav

Introducción

- Observe que $r \in h(p,q)$ si y sólo si $\|rp\| < \|rq\|$
- De esto se obtiene la siguiente observación:

$$\mathcal{V}(p_i) = \bigcap_{1 \leq j \leq n,\ j \neq i} h(p_i, p_j) \tag{2}$$

- Así pues, $\mathcal{V}(p_i)$ es la intersección de $n-1$ semiplanos, y por lo tanto una región poligonal convexa delimitada por a lo más $n-1$ vértices y como máximo $n-1$ aristas

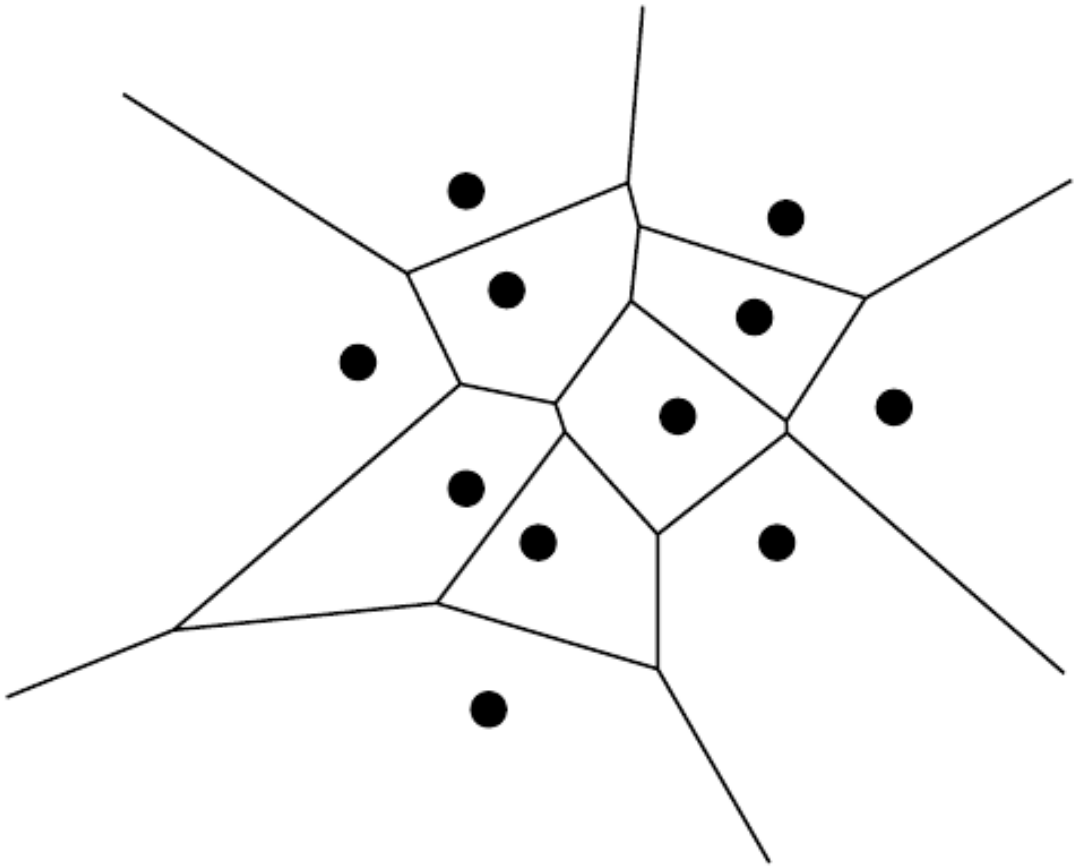
Cinvestav

Introducción

- Dado que la intersección de semiplanos es un polígono convexo posiblemente no acotado, es fácil ver que $\mathcal{V}(p_i)$ también lo es
- Por último, definamos el *diagrama de Voronoi* de P, denotado $Vor(P)$, como la subdivisión del plano en n celdas de Voronoi $\mathcal{V}(p_i)$, una para cada sitio en P
- No es difícil demostrar (ver libro de Berg et al.) que el diagrama de Voronoi se compone de una colección de segmentos de línea, que pueden ser no acotados, ya sea en un extremo o en ambos
- Un ejemplo se muestra en la siguiente figura

Cinvestav

Introducción



Cinvestav

Algunas aplicaciones

Búsqueda del vecino más cercano:

- Uno de los problemas más importantes en estructuras de datos para geometría computacional es la respuesta a búsquedas de vecinos más cercanos
- Dado un conjunto de puntos P, y un punto de consulta q, determinar el punto más cercano a q en P
- Esto puede responderse construyendo en primer lugar un diagrama de Voronoi para luego localizar la celda del diagrama que contiene a q

Cinvestav

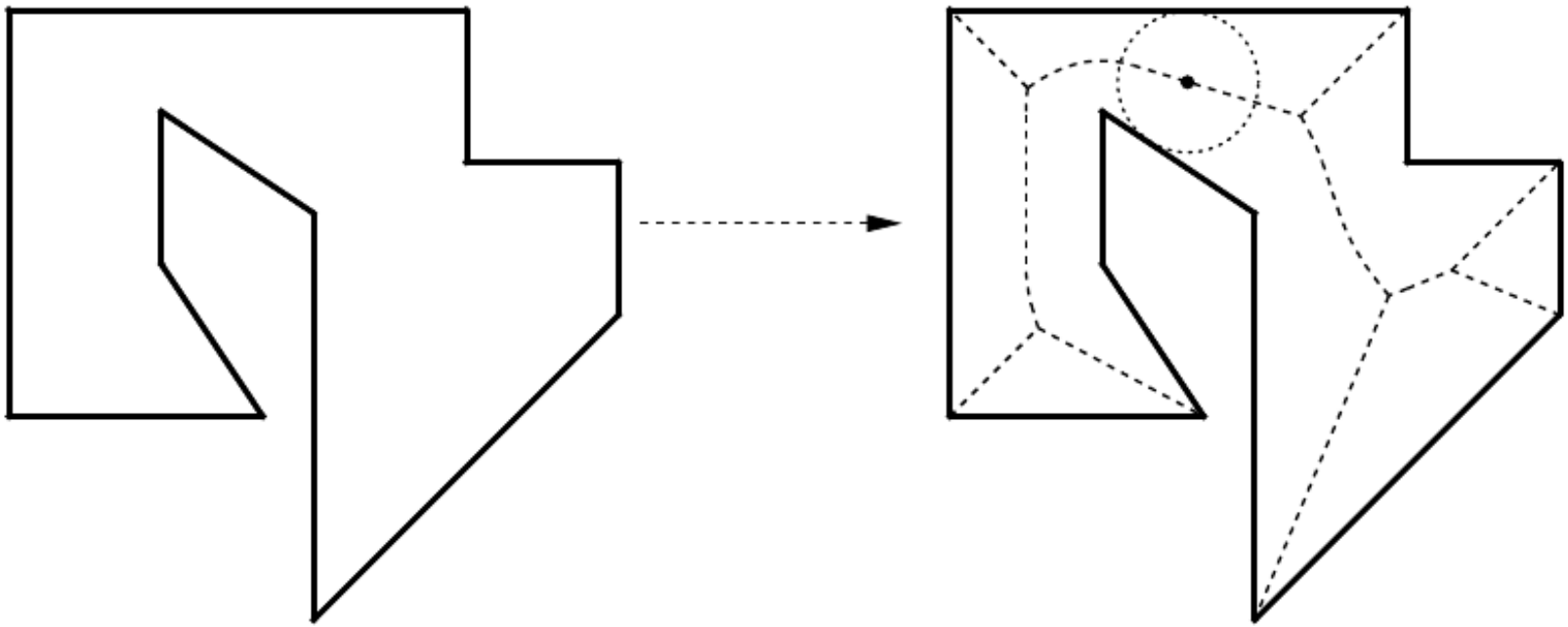
Algunas aplicaciones

Morfología computacional y análisis de forma:

- Una estructura útil en el análisis de forma es el eje medial
- El *eje medial* de una forma (por ejemplo, un polígono simple) se define como la unión de los puntos centrales de todos los discos localmente máximos que están contenidos dentro de la forma
- Si generalizamos la noción de diagrama de Voronoi para permitir que los sitios sean tanto puntos como segmentos de línea, entonces el eje medial de un polígono simple se puede extraer con facilidad a partir del diagrama de Voronoi de estos sitios generalizados

Cinvestav

Algunas aplicaciones



Algunas aplicaciones

Ubicación de instalaciones (*facility location*):

- Queremos abrir un nuevo Blockbuster
- Éste debe colocarse lo más lejos posible de cualquier otro vídeoclub existente
- ¿Dónde debe ser ubicado?
- Dado que los vértices del diagrama de Voronoi son los puntos que están localmente a máxima distancia de cualquier otro punto en el conjunto, entonces forman un conjunto de ubicaciones candidatas naturales

Cinvestav

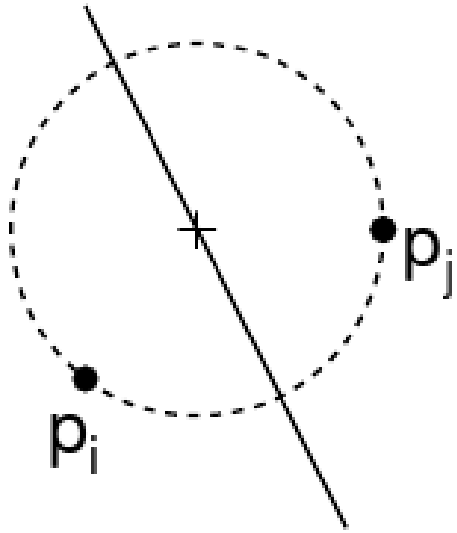
Cinvestav

Propiedades de los diagrama de Voronoi

- Veremos a continuación algunas propiedades sobre la estructura de los diagramas de Voronoi en el plano
- Evidentemente un diagrama de Voronoi es un conjunto de celdas cuyas caras son polígonos convexos (posiblemente no acotados)
- Cada punto en una *arista del diagrama de Voronoi* es equidistante de sus dos vecinos más cercanos p_i y p_j
- Por lo tanto, existe un círculo centrado en ese punto tal que p_i y p_j se encuentran en este círculo, y ningún otro sitio está al interior de él

Cinvestav

Propiedades de los diagrama de Voronoi

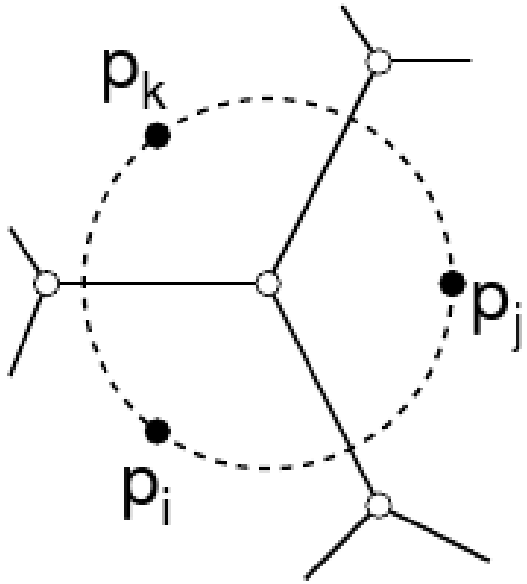


Propiedades de los diagrama de Voronoi

- El vértice en el cual tres celdas de Voronoi $\mathcal{V}(p_i)$, $\mathcal{V}(p_j)$, y $\mathcal{V}(p_k)$ se intersectan, es llamado un *vértice de Voronoi* y tiene la propiedad de ser equidistante de los sitios p_i, p_j, y p_k
- Por lo tanto, es el centro del círculo que pasa por estos sitios
- Además este círculo no contiene otros sitios en su interior

Cinvestav

Propiedades de los diagrama de Voronoi



Propiedades de los diagrama de Voronoi

- Generalmente tres puntos en el plano definen un único círculo
- Si hacemos la suposición general de que no existe ningún grupo de cuatro sitios cocirculares, entonces todos los vértices del diagrama de Voronoi tienen **grado** tres

Cinvestav

Propiedades de los diagrama de Voronoi

- Una celda del diagrama de Voronoi es no limitada si y sólo si el sitio correspondiente se encuentra en la **cubierta convexa**
- Observe que un sitio está en la cubierta convexa si y sólo si es el punto más cercano de algún punto en el infinito
- Así, dado un diagrama de Voronoi, es fácil extraer la cubierta convexa en tiempo lineal

Cinvestav

Propiedades de los diagrama de Voronoi

- Si n representa el número de sitios, entonces el diagrama de Voronoi es un grafo planar con exactamente n caras (imaginando que todas las aristas no acotadas estuvieran dirigidas a un vértice común infinito)
- Se desprende de la fórmula de Euler que el número de vértices de Voronoi es como máximo $2n - 5$ y el número de aristas es a lo más $3n - 6$ (ver demostración en el libro de Berg et al.)

Cinvestav

Cinvestav

Construcción de diagramas de Voronoi

- Existen una serie de algoritmos para construir el diagrama de Voronoi de un conjunto de n sitios en el plano
- Por supuesto, existe un algoritmo trivial que corre en tiempo $O(n^2 \log n)$, y que opera construyendo $\mathcal{V}(p_i)$ mediante la intersección de $n-1$ semiplanos $h(p_i, p_j)$, para $1 \leq j \leq n,\ j \neq i$
- Sin embargo, hay formas mucho más eficientes, que corren en tiempo $O(n \log n)$

Cinvestav

Construcción de diagramas de Voronoi

- Históricamente, se conocieron algoritmos para construir diagramas de Voronoi que corrían en tiempo $O(n^2)$ y que se basaban en construcciones incrementales
- Posteriormente se desarrollaron algoritmos $O(n \log n)$ basados en el método de divide y vencerás, pero que eran bastante complejos [Shamos and Hoey, 1975]
- Finalmente Steven Fortune (1987) inventó un algoritmo de barrido de plano para resolver este problema que es mucho más simple y corre también en tiempo $O(n \log n)$
- Es este algoritmo el que vamos a discutir hoy en clase

Cinvestav

Cinvestav

Algoritmo de Fortune

- Antes de comenzar con la explicación del algoritmo de Fortune es importante considerar el por qué este algoritmo no fue inventado antes
- De hecho, es bastante más complicado que cualquier otro algoritmo de barrido del plano que hemos visto hasta ahora
- Recordemos que la clave para cualquier algoritmo de barrido del plano es la capacidad para descubrir todos los eventos próximos de manera eficiente
- Por ejemplo, en el algoritmo de barrido del plano para encontrar la intersección de segmentos de línea, estas son descubiertas antes de que la línea de barrido llegue a ellas (son agregados como eventos futuros)

Cinvestav

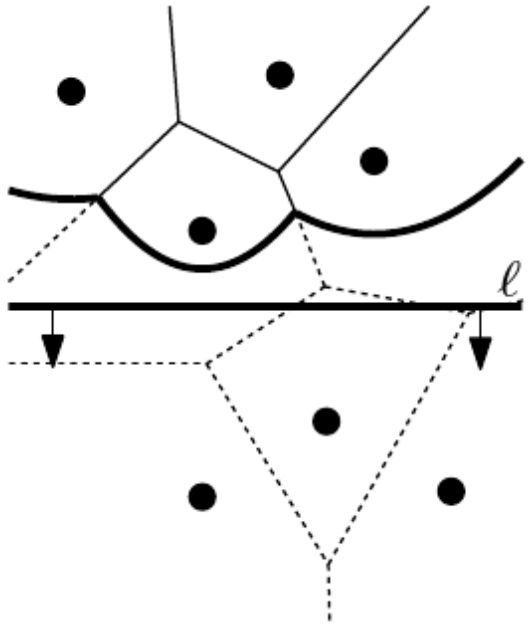
Algoritmo de Fortune

- El problema con el diagrama de Voronoi es el de predecir cuándo y dónde se producirán los próximos eventos
- Imagine que detrás de la línea de barrido ya se ha construido el diagrama de Voronoi basándose en los sitios que se han encontrado hasta ahora en el barrido
- La dificultad es que un sitio que queda por delante de la línea de barrido podría generar un vértice de Voronoi que se encuentre detrás de la línea de barrido

Cinvestav

Algoritmo de Fortune

- ¿Cómo puede saber el algoritmo de barrido de la existencia de este vértice antes de que pase por ese sitio?
- Son precisamente estos **eventos no anticipados** los que complican el diseño de un algoritmo de barrido del plano



Cinvestav

Algoritmo de Fortune

- Fortune hizo la inteligente observación de que podía calcularse el diagrama de Voronoi mediante barrido del plano construyendo una versión "distorsionada" de éste pero que es topológicamente equivalente
- Esta versión distorsionada del diagrama se basa en una transformación que modifica la forma en que las distancias son medidas en el plano
- El diagrama resultante tiene la misma estructura topológica que el diagrama de Voronoi, pero sus aristas son arcos parabólicos, en vez de segmentos de línea recta

Cinvestav

Algoritmo de Fortune

- Una vez que este diagrama distorsionado es obtenido, es fácil "corregirlo" en tiempo $O(n)$ para producir el diagrama de Voronoi correcto
- Nuestra presentación será diferente de la hecha originalmente por Fortune (y se basa en nuestro libro de texto)
- En vez de distorsionar el diagrama, podemos pensar en que este algoritmo realiza una alteración de la línea de barrido
- En realidad, vamos a utilizar dos objetos para controlar el proceso de barrido

Cinvestav

Algoritmo de Fortune

- En primer lugar, habrá una línea de barrido horizontal, moviéndose de arriba a abajo
- También vamos a mantener una curva x-monótona llamada **línea de playa** (beach line)
- A medida que la línea de barrido se mueve hacia abajo, la línea de playa va justo detrás siguiéndola

Cinvestav

Línea de playa

- Recordemos que el problema con el barrido del plano normal es que los sitios que se encuentran por debajo de la línea de barrido pueden afectar el diagrama que se encuentra por encima de la línea de barrido
- Para evitar este problema, se mantendrá sólo la parte del diagrama que no pueden ser afectada por cualquier cosa que se encuentra por debajo de la línea de barrido

Cinvestav

Línea de playa

- Para ello, vamos a subdividir el semiplano que cae por arriba de la línea de barrido en dos regiones:
  - Los puntos que están más cerca de algún sitio p por arriba de la línea de barrido, de lo que están de la misma línea de barrido
  - Los puntos que están más cerca de la línea de barrido que cualquier sitio arriba de ésta
- ¿Cuáles son las propiedades geométricas de la frontera entre estas dos regiones?
- El conjunto de puntos que se encuentran a la misma distancia de la línea de barrido y de su sitio más cercano por encima de la línea de barrido es lo que denominamos **línea de la playa**

Cinvestav

Línea de playa

- Observe que, para cualquier punto q por encima de la línea de playa, sabemos que su sitio más cercano no pueden ser afectado por ningún sitio que se encuentra por debajo de la línea de barrido
- Por lo tanto, la parte del diagrama de Voronoi que se encuentra por encima de la línea de playa está "segura"
- Esto en el sentido de que tenemos toda la información que necesitamos para calcularla (sin necesidad de conocer qué sitios todavía aparecen por debajo de la línea de barrido)

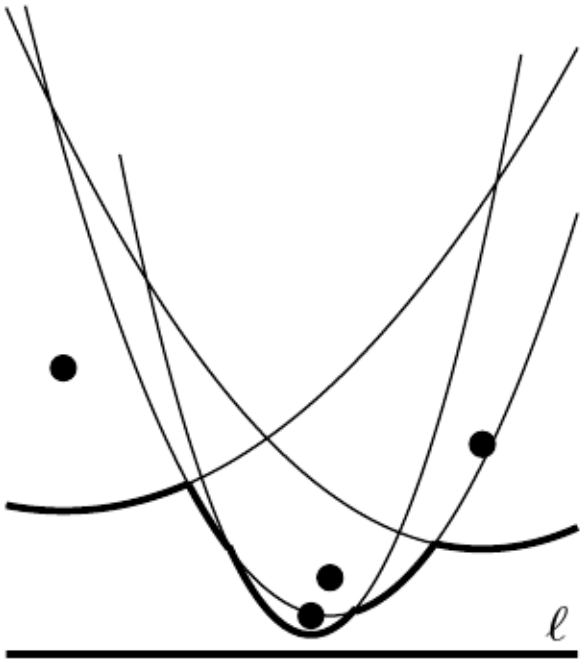
Cinvestav

Línea de playa

- ¿Qué forma tiene la línea de playa?
- Recordemos de la clase de geometría en la secundaria que el conjunto de puntos que están equidistantes de un punto (un sitio) y de una línea (la línea de barrido) forman una parábola
- Es evidente que la forma de la parábola cambia continuamente conforme la línea de barrido se mueve
- Con un poco de geometría analítica, es fácil demostrar que la parábola se hace menos ancha cuando el sitio está más cerca de la línea y más ancha conforme la línea de barrido se aleja

Cinvestav

Línea de playa



Línea de playa

- Por lo tanto, la línea de playa consiste de la *envoltura inferior* de estas parábolas, una para cada sitio
- Tenga en cuenta que la parábola de algunos sitios por encima de la línea de playa no tocará la envoltura inferior, y por lo tanto no contribuirá a la línea de playa
- Debido a que las parábolas son x-monótonas, la línea de playa también lo es

Cinvestav

Línea de playa

- También observe que el punto donde dos arcos de la línea de playa se intersectan, el cual llamamos *punto de corte*, es equidistante de dos sitios y de la línea de barrido, por lo tanto deberá situarse en alguna arista del diagrama de Voronoi
- En particular, si los arcos de la línea de playa correspondientes a los sitios p_i y p_j comparten un punto de corte en común en la línea de playa, este punto de corte caerá en la arista de Voronoi entre p_i y p_j

Cinvestav

Línea de playa

- Como todos los algoritmos de barrido del plano, el algoritmo de Fortune mantiene un registro del estatus de la línea de barrido así como de ciertos eventos significativos
- En este caso cualquier evento que cambia la estructura topológica del diagrama de Voronoi y de la línea de playa
- En el **estatus de la línea de barrido** el algoritmo mantiene la posición actual (coordenada y) de la línea de barrido
- Además almacena en orden de izquierda a derecha la secuencia de sitios que definen la línea de playa (no almacena las parábolas)

Cinvestav

Línea de playa

- Por otra parte el algoritmo de Fortune trabaja con 2 tipos de eventos:
  - **Eventos de sitio:** Cuando la línea de barrido pasa sobre un nuevo sitio un nuevo arco será insertado en la línea de playa
  - **Eventos de vértices de Voronoi:** (llamados eventos círculo en el texto) Cuando la longitud de un arco parabólico se reduce a cero, el arco desaparece y un nuevo vértice de Voronoi será creado en ese punto

Cinvestav

Línea de playa

- El algoritmo consiste en el tratamiento de estos dos tipos de eventos
- A medida que los vértices de Voronoi son descubiertos con *eventos de vértices de Voronoi*, será fácil actualizar el diagrama (suponiendo una representación adecuada de las celdas), y unir todo el diagrama
- Veamos más a detalle los dos tipos de eventos que utiliza el algoritmo de Fortune

Cinvestav

Eventos de sitio

- Un evento de sitio se genera cuando la línea de barrido pasa sobre un sitio p_i
- Como mencionamos antes, en el instante en que la línea de barrido toca el punto, su arco parabólico asociado degenera en un rayo vertical que sale hacia arriba del punto en dirección de la línea de playa actual
- Conforme la línea de barrido baja, este rayo va ampliándose en un arco a lo largo de la línea de playa
- Para procesar un *evento de sitio* se determinará el arco de la línea de barrido que se encuentra directamente sobre el nuevo sitio (hagamos la suposición general de que no está inmediatamente abajo de un vértice de la línea de playa)

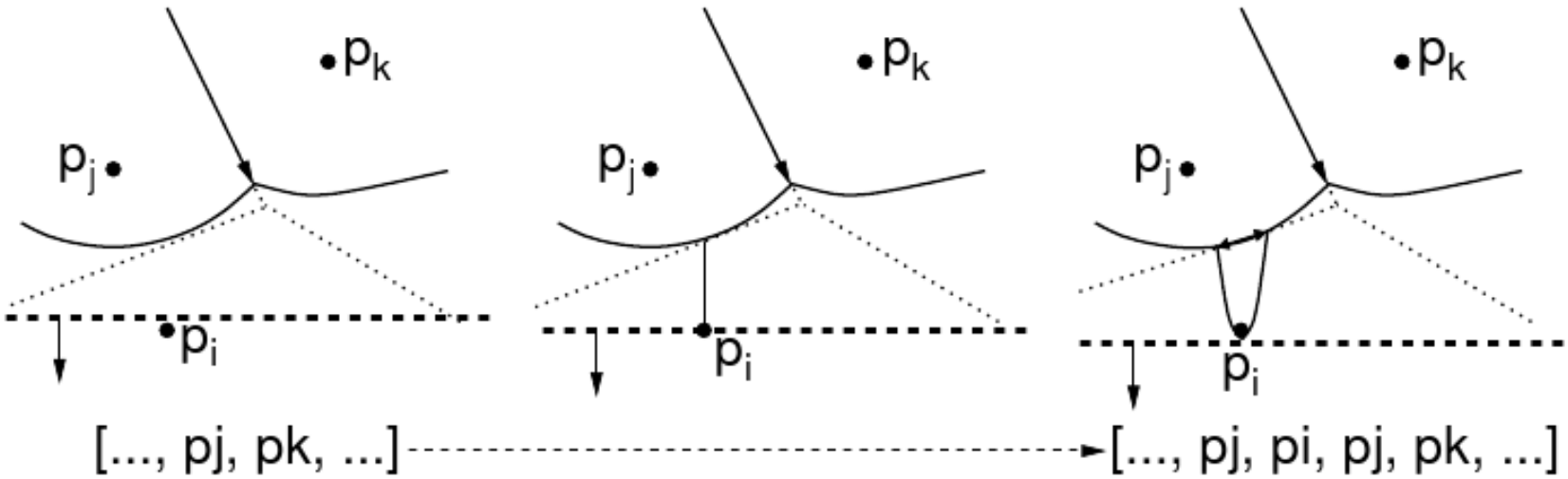
Cinvestav

Eventos de sitio

- Sea p_j el sitio que genera este arco
- A continuación, dividiremos este arco de la línea de playa en dos mediante la inserción de una nueva entrada para este punto en el estatus de la línea de barrido
- Inicialmente éste corresponde a un arco infinitesimalmente pequeño a lo largo de la línea de playa, pero conforme la línea de barrido avanza, este arco se volverá más amplio
- Por lo tanto, la entrada $\langle \ldots, p_j, \ldots \rangle$ en el estatus de la línea de barrido se sustituirá por la tripleta $\langle \ldots, p_j, p_i, p_j \ldots \rangle$

Cinvestav

Eventos de sitio



Cinvestav

Eventos de sitio

- Es importante considerar que esta es la única manera en la que nuevos arcos pueden ser introducidos en la línea de playa
- No vamos a probarlo en clase, pero es recomendable que estudie la demostración presentada en el libro de texto
- Ya que como consecuencia de ésta, se desprende que el número de arcos en la línea de playa puede ser como máximo $2n - 1$, ya que cada nuevo punto puede resultar en la creación de un nuevo arco, y la división de un arco existente, para un aumento neto de dos arcos por punto (excepto para el primero)

Cinvestav

Eventos de sitio

- Tenga en cuenta que un punto puede contribuir en general más de un arco a la línea de playa
- La ventaja de los eventos de sitio es que todos ellos son conocidos de antemano
- Así, después de ordenar los puntos de acuerdo a sus coordenadas y, todos estos eventos son conocidos

Cinvestav

Eventos de vértices de Voronoi

- En contraste con los eventos del sitio, los eventos de vértices de Voronoi son generados dinámicamente conforme el algoritmo se ejecuta
- Al igual que en algoritmo de barrido del plano para intersección de segmentos de línea, la idea clave aqui es que cada uno de esos eventos sea generado por objetos que son *vecinos* en la línea de playa
- Sin embargo, a diferencia del algoritmo para la intersección de segmentos donde pares de segmentos consecutivos generan eventos, aquí son tripletas de puntos las que generan los nuevos eventos

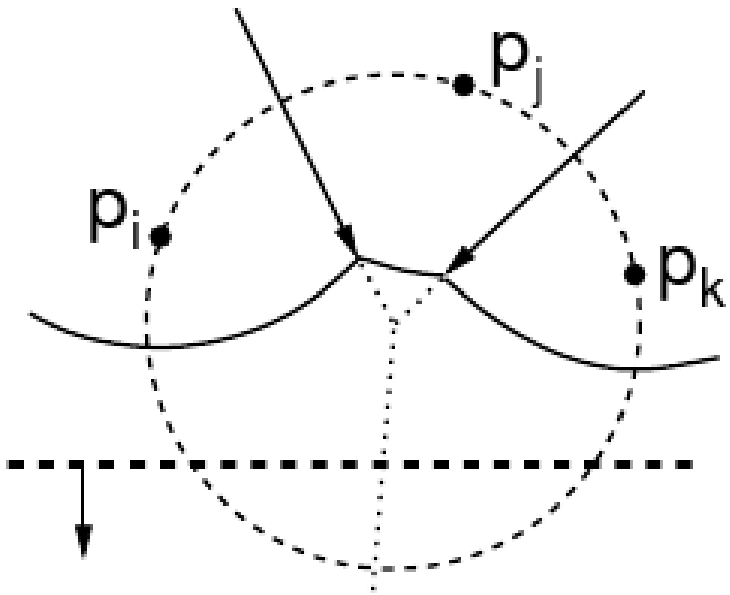
Cinvestav

Eventos de vértices de Voronoi

- En particular, consideremos cualquier tripleta de sitios p_i, p_j, y p_k cuyos arcos aparecen consecutivamente en la línea de playa de izquierda a derecha (ver figura)
- Por otra parte, supongamos que la circunferencia circunscrita de estos tres sitios cae al menos parcialmente por debajo de la línea de barrido actual (lo que significa que el vértice Voronoi aún no se ha generado)
- Además de que ésta no contiene puntos que están situados por debajo de la línea de barrido (lo que significa que no hay ningún punto futuro que bloqueará la creación del vértice)

Cinvestav

Eventos de vértices de Voronoi



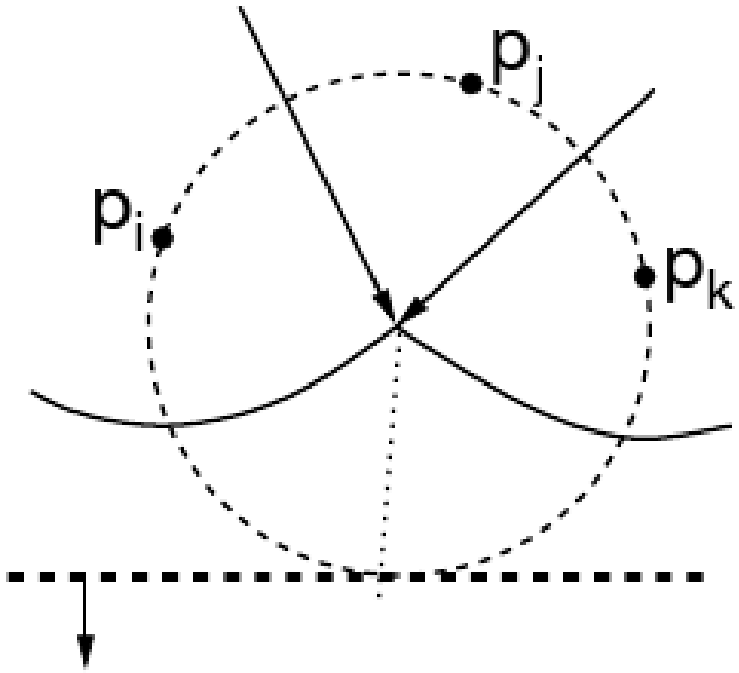
[..., pi, pj, pk, ...]

Eventos de vértices de Voronoi

- Considere el momento en que la línea de barrido llega a un punto en el que es tangente al punto más bajo de este círculo
- En este instante el centro de la circunferencia circunscrita es equidistante de los tres sitios y de la línea de barrido
- Por lo que los tres arcos parabólicos pasan por este punto central, lo que implica que la contribución del arco de p_j ha desaparecido de la línea de playa

Cinvestav

Eventos de vértices de Voronoi

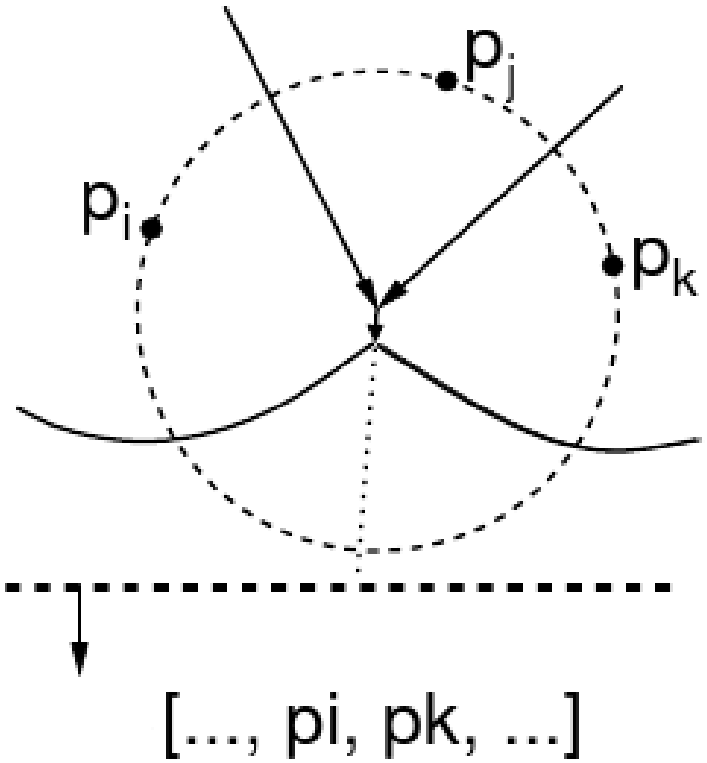


Eventos de vértices de Voronoi

- En términos del diagrama de Voronoi, las mediatrices (p_i, p_j) y (p_j, p_k) se han unido en un vértice de Voronoi
- Quedando una sola mediatriz (p_i, p_k)
- Por lo tanto, la tripleta de sitios consecutivos p_i, p_j, y p_k en el estatus de la línea de barrido es remplazada por p_i, p_k

Cinvestav

Eventos de vértices de Voronoi



Algoritmo de barrido del plano

- Ahora podemos presentar el algoritmo de barrido del plano con más detalle
- Las principales estructuras que vamos a mantener son las siguientes:
  - Diagrama de Voronoi (parcial)
  - Línea de playa
  - Cola de eventos

Cinvestav

Algoritmo de barrido del plano

Diagrama de Voronoi parcial:

- El diagrama de Voronoi parcial que ha sido construido hasta el momento será conservado en cualquier estructura de datos adecuada para almacenar subdivisiones planares, por ejemplo, una lista de aristas doblemente conectada
- Hay un problema técnico causado por el hecho de que el diagrama contiene aristas no acotadas
- Para manejar esto vamos a suponer que todo el diagrama está situado dentro de un rectángulo (*bounding box*).
- Este rectángulo debe elegirse suficientemente grande para que todos los vértices de Voronoi queden dentro de él

Cinvestav

Algoritmo de barrido del plano

Línea de playa:

- La línea de playa consiste de una secuencia ordenada de los sitios cuyos arcos parabólicos la forman
- Es representada mediante un diccionario, por ejemplo un árbol binario balanceado o una lista por saltos (*skip list*)
- Como se ha mencionado anteriormente, no se almacenan explícitamente los arcos parabólicos
- Esos arcos sólo existen para fines de derivar el algoritmo

Cinvestav

Algoritmo de barrido del plano

- En lugar de almacenar cada arco parabólico sobre la actual línea de playa, guardamos únicamente el sitio que da lugar a este arco
- La operación clave de búsqueda es aquella que permite ubicar el arco de la línea de playa que se encuentra directamente arriba de un sitio recientemente descubierto
- Entre cada par consecutivo de sitios p_i y p_j, hay un *punto de corte*
- A pesar de que el punto de corte se mueve en función de la línea de barrido, observemos que es posible calcular la ubicación exacta de éste en función de p_i, p_j, y la coordenada y actual de la línea de barrido

Cinvestav

Algoritmo de barrido del plano

- En particular, el punto de corte es el centro de un círculo que pasa a través de p_i, p_j y es tangente a la línea de barrido
- Por lo tanto, como con la línea de playa, no almacenamos explícitamente los puntos de corte
- En lugar de esto, se calculan sólo cuando se necesitan

Cinvestav

Algoritmo de barrido del plano

- Una vez que el punto de corte se calcula, entonces podemos determinar si un sitio agregado recientemente está a su izquierda o a su derecha
- Usando un ordenamiento de los sitios y primitivas de comparación se puede conducir una búsqueda binaria para encontrar el arco que cae por arriba del nuevo sitio
- Las operaciones que serviran para mantener la línea de playa son las siguientes:

Cinvestav

Algoritmo de barrido del plano

Búsqueda:

- Dada la coordenada y actual de la línea de barrido y un nuevo sitio p_i, determinar el arco de la línea de playa que se encuentra justo arriba de p_i
- Denotemos p_j el sitio que contribuye con este arco
- Finalmente se devuelve una referencia a esta entrada de la línea de playa

Cinvestav

Algoritmo de barrido del plano

Inserción y división:

- Se inserta una nueva entrada para p_i dentro de un arco p_j de la línea de playa
- Por lo tanto, se hace la sustitución de un arco simple $\langle \ldots, p_j, \ldots \rangle$ por la tripleta $\langle \ldots, p_j, p_i, p_j \ldots \rangle$
- Se regresa una referencia a la nueva entrada de la línea de playa para su futuro uso

Cinvestav

Algoritmo de barrido del plano

Eliminación:

- Dada la referencia a una entrada p_j de la línea de playa, se elimina ésta
- Esto sustituye una tripleta $\langle \ldots, p_i, p_j, p_k \ldots \rangle$ con el par $\langle \ldots, p_i, p_k \ldots \rangle$
- Como podemos ver no sería difícil modificar una estructura de datos estándar para realizar cada una de estas operaciones en un tiempo $O(\log n)$

Cinvestav

Algoritmo de barrido del plano

Cola de eventos:

- La cola de eventos es una cola por prioridad con la capacidad para insertar y eliminar los nuevos eventos
- También el evento con la mayor coordenada y debe poder ser extraído
- Para cada sitio almacenamos su coordenada y en la cola
- Todas las operaciones pueden ser implementadas para ejecutarse en tiempo $O(\log n)$ asumiendo que la cola de prioridad se almacena como un diccionario ordenado

Cinvestav

Algoritmo de barrido del plano

- Para cada tripleta consecutiva p_i, p_j, p_k en la línea de playa, calculamos la circunferencia circunscrita de estos puntos, lo cual se puede hacer en tiempo $O(1)$
- Si el punto extremo inferior del círculo (mínima coordenada y) se encuentra por debajo de la línea de barrido, entonces se crea un evento de vértice de Voronoi cuya coordenada y es la misma que la de ese punto
- Se almacena este evento en la cola, donde cada uno de esos eventos tiene un apuntador a la tripleta de sitios que lo generó
- Además cada tripleta consecutiva de sitios cuenta con un apuntador hacia al evento que ésta generó en la cola de eventos

Cinvestav

Algoritmo de barrido del plano

- El algoritmo procede como cualquier algoritmo de barrido del plano
- Este inicia con la inserción del vértice extremo superior en el estatus de la línea de barrido
- Despúes se extrae un evento, se procesa, y se pasa al siguiente evento

Cinvestav

Algoritmo de barrido del plano

- Cada evento puede resultar en una modificación del diagrama de Voronoi y de la línea de playa
- Además también puede resultar en la creación o supresión de los eventos existentes
- Ahora veremos más a detalle como se procesan los dos tipos de eventos con los que trabaja el algoritmo

Cinvestav

Algoritmo de barrido del plano

Eventos de sitio: Sea p_i el nuevo sitio

1. Avanza la línea de barrido ℓ de manera que pase sobre p_i. Buscar el arco de la línea de playa que cae inmediatamente arriba de p_i, sea p_j el sitio correspondiente a ese arco
2. Aplicando la operación de inserción y división, se agrega una nueva entrada para p_i, sustituyendo así $\langle \ldots, p_j, \ldots \rangle$ por $\langle \ldots, p_j, p_i, p_j \ldots \rangle$
3. Se crear una nueva arista en el diagrama de Voronoi, la cual se encuentra en la bisectriz entre p_i y p_j
4. Algunas tripletas antiguas conteniendo p_j podrían ser suprimidas y otras nuevas con p_i insertadas, dependiendo del cambio de vecinos sobre la línea de playa. La tripleta creada p_j, p_i, p_j no puede generar un evento porque implica sólo dos sitios distintos

Cinvestav

Algoritmo de barrido del plano

Eventos de vértices de Voronoi: Sean p_i, p_j y p_k los sitios que generaron este evento (de izq. a derecha)

1. Borrar la entrada para p_j de la línea de la playa (por lo tanto se elimina su arco)
2. Crear un nuevo vértice en el diagrama de Voronoi (en el circuncentro de $\{p_i, p_j, p_k\}$), y unir las dos aristas de Voronoi de las mediatrices $(p_i, p_j), (p_j, p_k)$ a este vértice (estas aristas se generaron durante el evento de sitio paso 3 del algoritmo anterior)
3. Crear una nueva arista para la bisectriz entre p_i y p_k
4. Eliminar cualquier evento que surgió de las tripletas que contienen el arco de p_j, y generar los nuevos eventos correspondientes a las tripletas consecutivas con p_i y p_k

Cinvestav

Cinvestav

Análisis de complejidad

- Cada evento implica un tiempo de procesamiento $O(1)$ más un número constante de operaciones a las distintas estructuras de datos
- El tamaño de las estructuras de datos es $O(n)$, y cada una de estas operaciones consume un tiempo $O(\log n)$
- Por lo tanto, el tiempo total es $O(n \log n)$, y el espacio total es $O(n)$

Cinvestav