Triangulaciones y mallas de polígonos: modelación en ingeniería y CG

María Cecilia Rivara
mcrivara@dcc.uchile.cl
Universidad de Chile
2016

Bosquejo

- Aplicaciones simples de terrenos
  - Mallas uniformes (datos de satélites)
  - Terrenos fractales (síntesis de terrenos)
- Triangulaciones 2½D para terrenos
- Curvas de nivel sobre triangulaciones 2½D
- Mallas de polígonos, triangulaciones en ingeniería
- Triangulación de Delaunay y algoritmo para construir triangulaciones

Modelación simple de terrenos mediante mallas uniformes

- Supuesto: terreno es función $h(x,y)$ en el espacio 3D, con valor único $h(x,y)$ para cada (x,y). Definido sobre un rectángulo

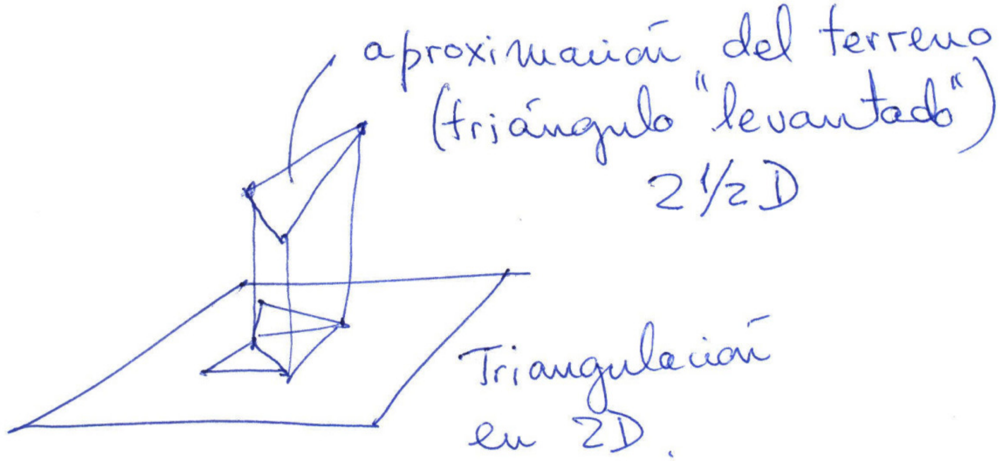
- Se discretiza el rectángulo en malla uniforme (o no uniforme) de polígonos en 2D (cuadriláteros o triángulos) en plano x,y.

- A cada punto discretizado x,y se asocia su elevación $h(x,y)$. Se dice que se tiene triangulación / malla en 2½ D.

Triangulaciones / mallas en 2 ½ D

- En general en aplicaciones de terrenos se usan mallas 2½D.
- Estas técnicas no sirven para triangularizar la superficie de objetos complejos.

Aproximación del terreno



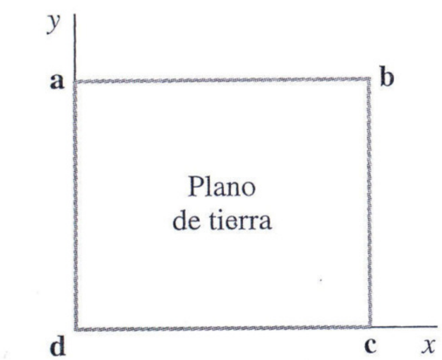
Fractales

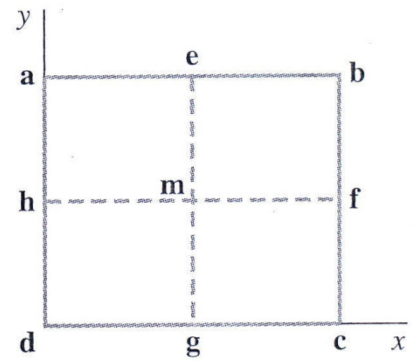
- Fractales: permiten construir excelentes imágenes sintéticas (artificiales) de fenómenos naturales (copos de nieve, árboles, nubes, terrenos)
- Geometría fractal: a distintas escalas se repite la misma geometría
  - Fractales deterministas autosimilares
  - Fractales estadísticamente autosimilares (con componente aleatorio)

Terrenos "fractales"

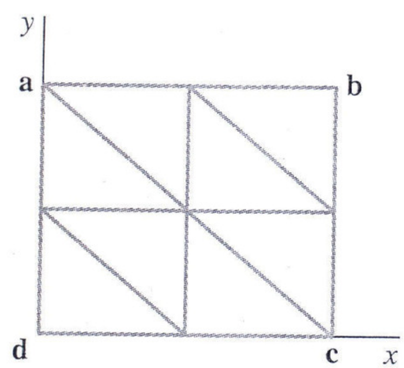
- Procedimientos computacionales simples para generar terrenos.
- Aplicaciones de juegos y CG
- Metodología: división de polígonos (triángulos / cuadriláteros) por el punto medio y asignación de elevación con componente aleatorio.
- Se obtienen mallas uniformes en 2D

Inicio de cálculo para obtener terreno fractal

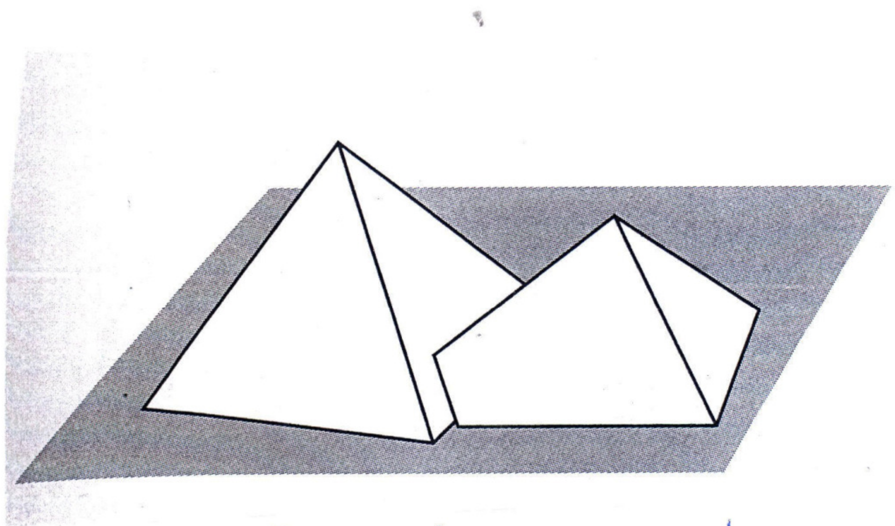




Inicios alternativos
terreno fractal



Terreno simple de inicio



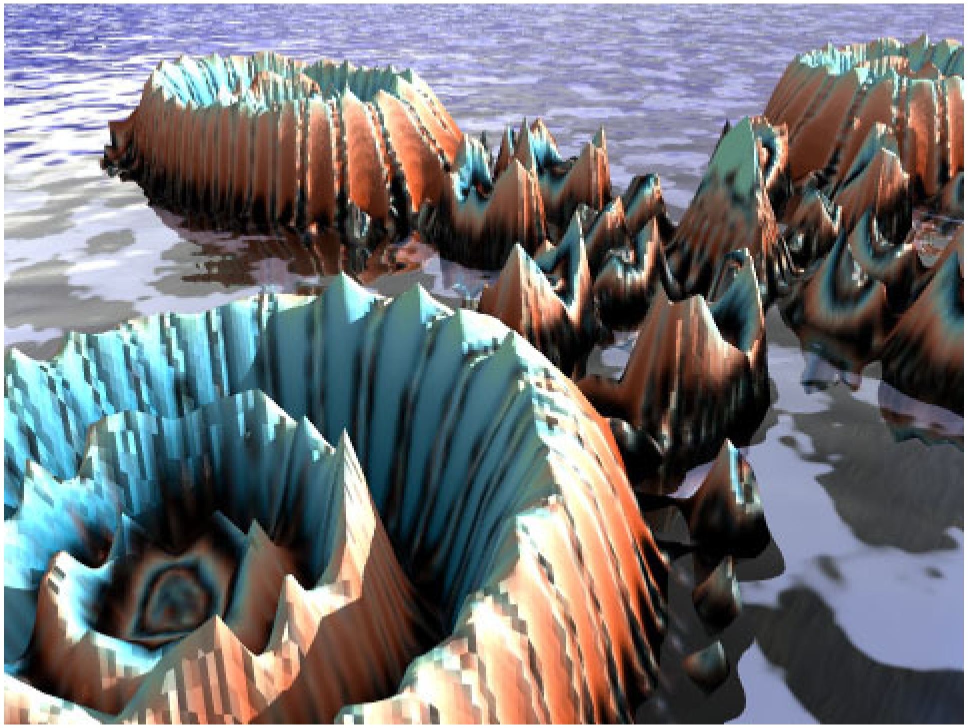
Superficie de inicio terreno fractal

Método

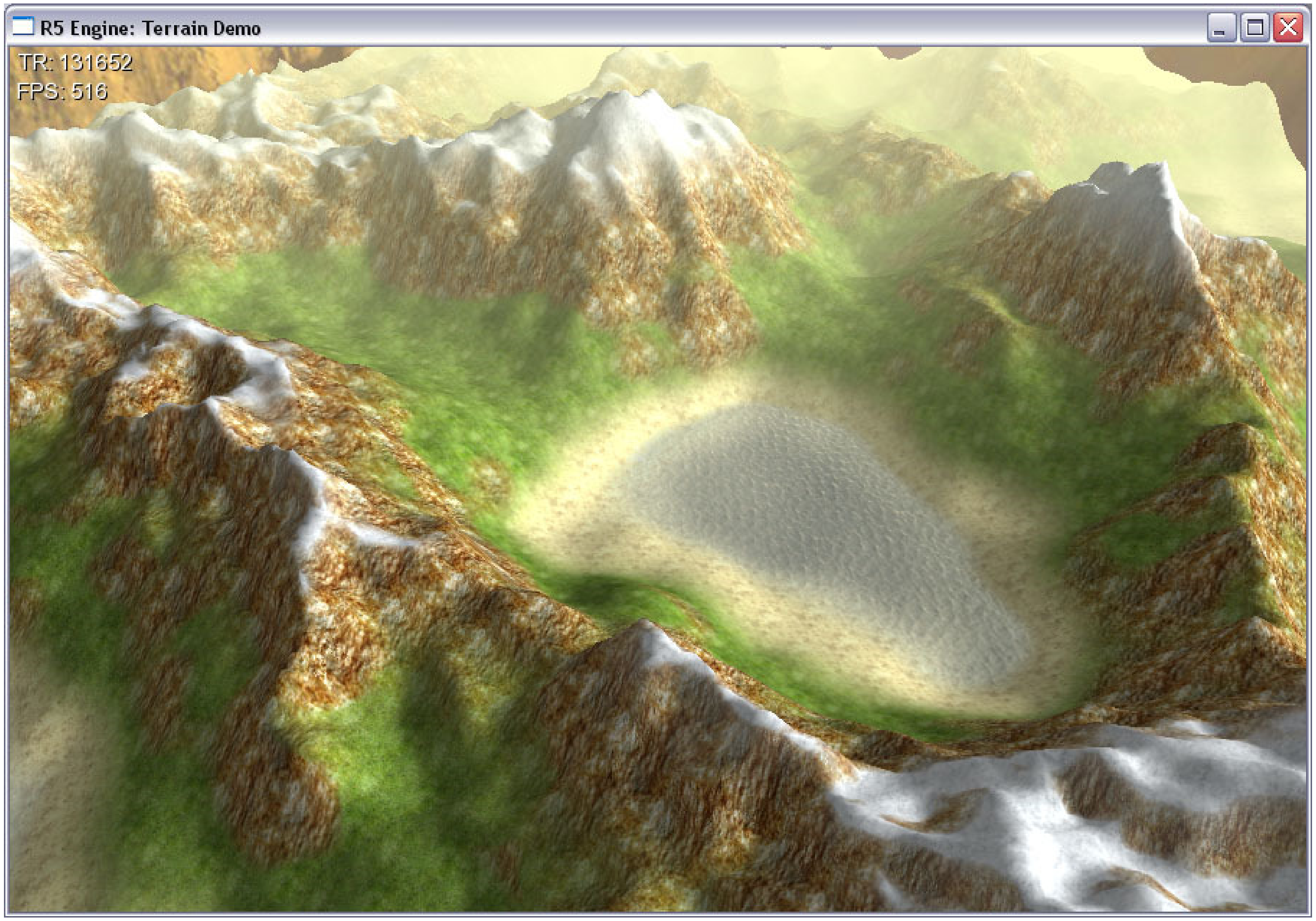
- Se divide cada polígono por el punto medio

- Para cada punto se encuentra elevación de inicio y se suma una componente aleatoria

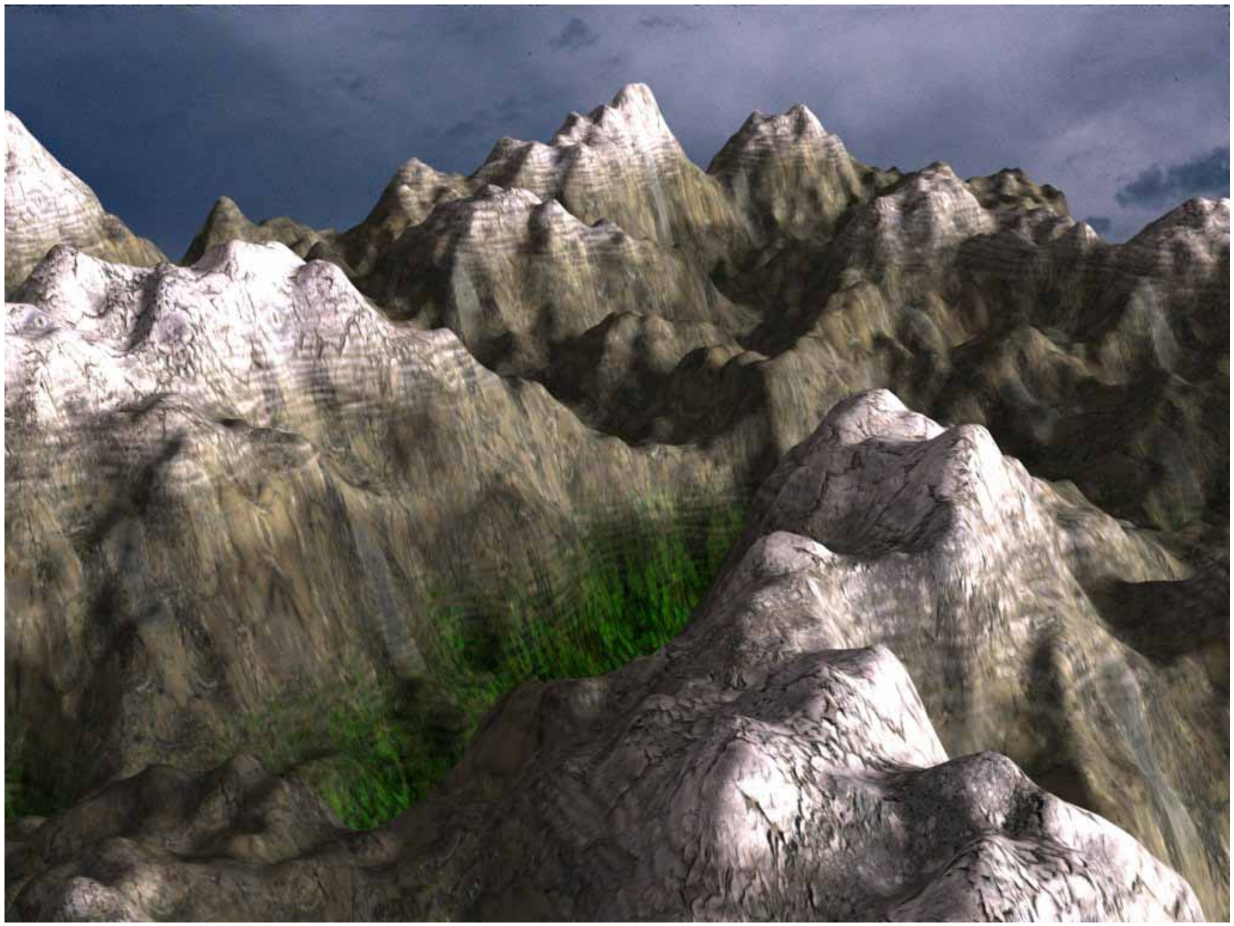
Terreno fractal



Terreno fractal



Terreno fractal

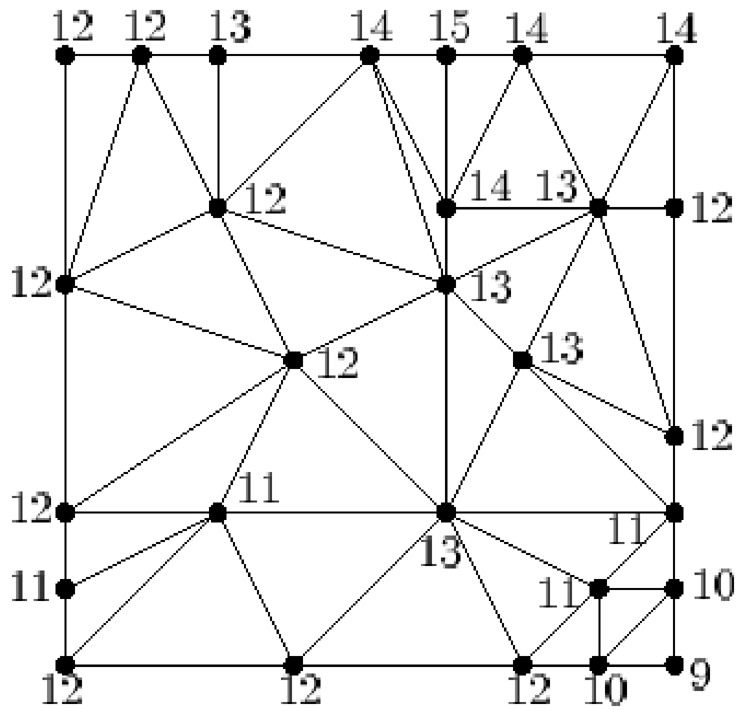


Construcción de curvas de nivel a partir de la triangulación en 2 ½ D

- Basta trabajar en 2D: se encuentran directamente las curvas de nivel proyectadas en plano x y.
- Curvas de nivel
  - Se aproximan por segmentos de líneas rectas sobre cada triángulo 2D
- Importante que la ED maneje información de vecindad entre triángulos.

Encontrar curva de nivel de elevación H =cte

- Encontrar un triángulo t que corte la curva
- Encontrar aristas de t que cortan la curva (vértices con elevación mayor y menor que H). Calcular intersección con las aristas y unir estos puntos.
- Seguir por triángulo vecino a arista que corta curva de nivel


12
12
13
14
15
14
14
12
14
13
12
12
13
13
12
12
12
11
11
13
11
11
10
12
12
12
10
9

Mallas de polígonos y triangulaciones generales en aplicaciones de ingeniería y ciencias

Observación

- En aplicaciones “blandas” de computación gráfica (entretenimientos, juegos, etc) las superficies se modelan como conjuntos de polígonos / triángulos. Con normales asociadas a los vértices. En general se aceptan “errores” en las triangulaciones.
- En aplicaciones “duras” de ingeniería y ciencias necesitamos modelación precisa de los objetos geométricos (superficies, polígonos, poliedros, objetos 3D).

Malla de polígonos (I)

- Polígono: figura plana, finita y cerrada, limitada por segmentos de líneas rectas.
- Malla de polígonos M en 2D o 3D (en abstracto)

es un conjunto de polígonos $\{P_i\}_{i=1, N}$ tal que:

i) área $(P_i) > 0$ para todo i

ii) interior $(P_i) \cap$ interior$(P_j) = \emptyset$ para todo par i, j

iii) para cada par P_i, P_j de polígonos vecinos, $P_i \cap P_j$ es una arista común o un vértice común

iv) M define una superficie conexa (abierta o cerrada)

M es válida o conforme si cumple (ii) y (iii). Importante para validar mallas

Malla de polígonos (II)

Importante

- Malla de polígonos es un conjunto de polígonos con topología asociada
- Elementos topológicos: vértices, aristas, caras*
- Elementos geométricos: puntos, segmentos de línea recta y polígonos*
- Topología asociada: Relaciones de vencidad entre elementos topológicos
- M es válida o conforme si cumple (ii) y (iii). Importante para validar mallas

(*) Nota: abuso de lenguaje: cara = polígono

Representación / modelación rigurosa de mallas de polígonos y triangulaciones

- Las aplicaciones requieren de información sobre los elementos topológicos que componen las mallas.
- Es necesario diseñar estructuras de datos / representaciones adecuadas para las aplicaciones específicas.

Malla de polígonos M (III)

- Elemento fundamental: el polígono

  Debe aparecer de manera implícita o explícita en la estructura de datos / representación computacional de M.

- Ejemplo: mallas de polígonos en computación gráfica. Se necesita el polígono para calcular normales y poder usar el modelo de iluminación.

Malla de polígonos M (IV)

Estructuras de datos (dependen de las aplicaciones y sus necesidades)

- Basadas en polígonos (representaciones minimales)
- Basadas en aristas
  - Winged Edge Data Structure

    1973, 1975 (desarrollada para área CAD)
  - Variaciones
  - Half-edge data stracture
  - Otras

Ejemplos: estructuras de datos para triangulaciones

1. Representación ingenua

   vértice → coordenadas

   triángulo → $pV_1\ pV_2\ pV_3$

   pV_i : puntero a vértice V_i

No hay información de vecindad entre elementos topológicos

Otro ejemplo

2. Representación usada en aplicaciones del método de Elementos Finitos, método numérico para analizar fenómenos físicos modelados para EDPs.

Vértice → coordenadas

triángulo → pV_1 pV_2 pV_3

→ pt_1 pt_2 pt_3

pt_i: puntero a triángulo vecino por una arista.

Problemas cuyo objetivo es construir mallas de polígonos

- Datos
  - (D1) Conjuntos de punto $\{p_j\}_{j=1, n}$ en el plano
  - (D2) Dominio: superficie finita y cerrada en el plano o en el espacio
  - (D3) Conjuntos de puntos seleccionados de un dominio en IR^2 o superficie en IR^3
- Se busca
  - Encontrar malla de polígonos que 'relacione' los puntos de (D1).
  - Encontrar malla de polígonos que represente de manera exacta y discretizada el dominio (ejemplo: dominio poligonal en IR^2)
  - Encontrar una malla de polígonos que aproxime el dominio (ejemplos: dominio plano con bordes curvos, superficie curva en el espacio)

Construcción de triangulaciones

Triangulaciones (mallas de triángulos

- polígono = triángulo
- aproximan bien geometrías complejas
- no estructuradas (más difícil de obtener y manejar que mallas de cuadriláteros uniformes)
- flexibles y generales: se adaptan bien a las necesidades de aplicaciones complejas
- representaciones / estructuras de datos.
  - basadas en triángulos
  - basadas en aristas (half-edge DS, otras)
- requerimientos importantes
  - triangulaciones conformes
  - triángulos de buena calidad geométrica

Problemas / Algoritmos sobre triangulaciones (1)

- análisis mediante el método de elementos finitos clásico para EDPs
  - requiere discretización adecuada de la geometría
  - discretización que represente bien la solución de las EDPs.
- métodos de elementos finitos adaptivos
  - construyen automáticamente la triangulación adaptándose a la solución de la EDP

Problemas / Algoritmos sobre triangulaciones (2)

- problemas evolutivos que requieren que la malla cambie a través del tiempo.
- problemas con bordes / geometrías cambiantes
- aplicaciones de computación gráfica
  - simplificación de mallas enormes obtenidas por escaneo, etc.
- modelación de terrenos
- aproximación de superficies complejas

Problemas básicos (no simples) de triangulación

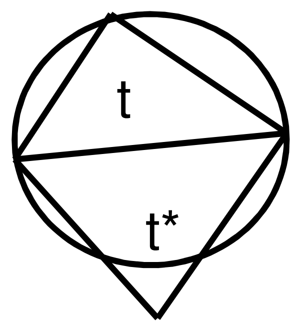
- Triangulación de conjuntos de puntos en un plano
- Triangulación de superficies / terrenos
- Triangulación de polígonos, PSLG
- Triangulaciones de puntos en 3D: mallas de tetraedros
- Triangulación 3D de geometrías complejas.

Delaunay triangulation (DT)

Problem: Given a point set P, construct a triangulation of P

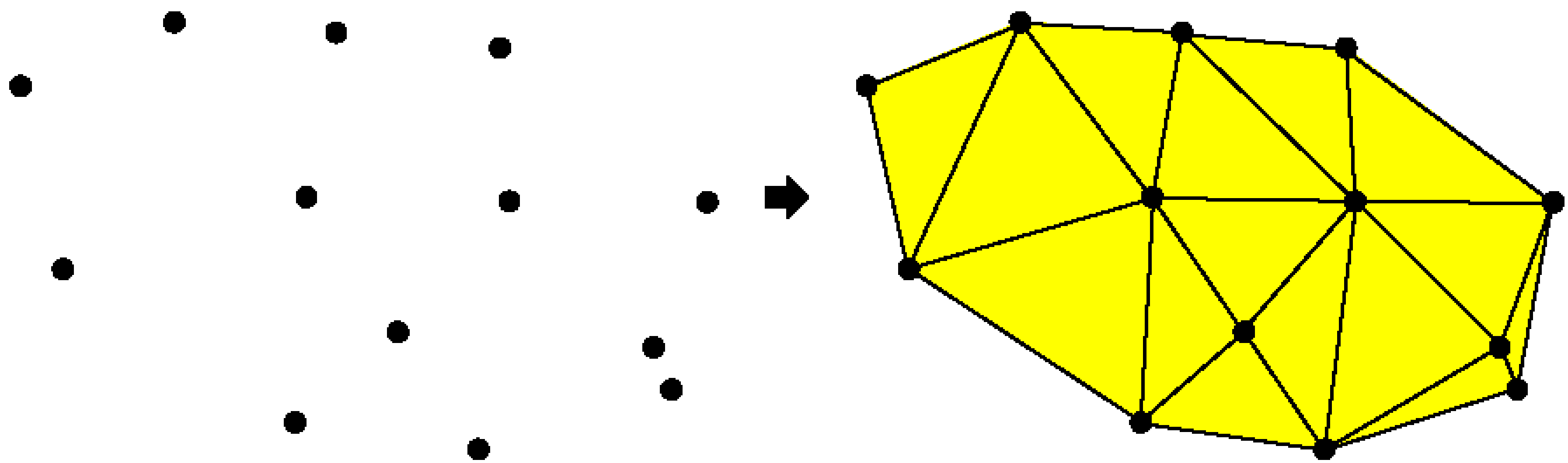
P = { points }
T is a DT of P if the circuncircle of every triangle t in T does not include any vertex of P in its interior

t and t* are locally Delaunay triangles

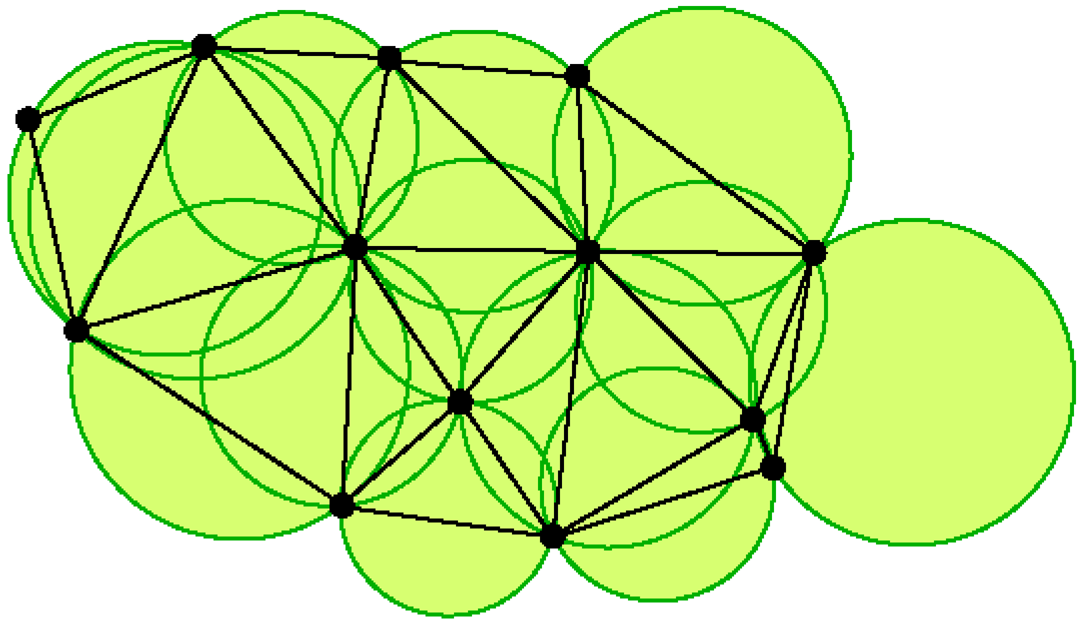


Remark: DT maximizes the minimun angle

Datos (puntos) y triangulación de Delaunay



T. Delaunay y circuncírculos

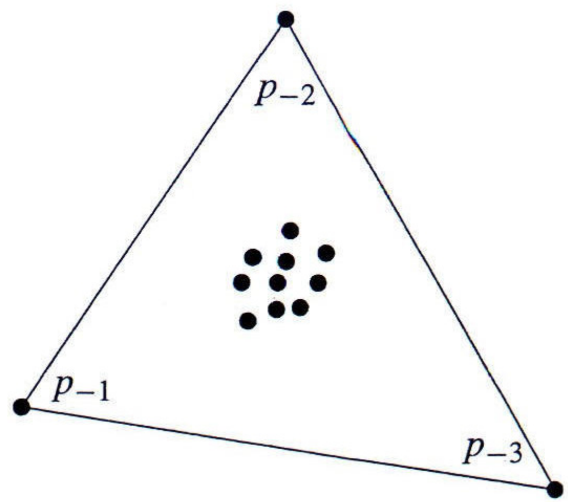


Propiedades de la Triangulación Delaunay

- Maximiza el ángulo mínimo: construye la triangulación más equilátera posible.
- Encuentra el cierre convexo de los puntos

Algoritmo más usado es incremental: se introducen los puntos uno a uno en triangulación de Delaunay previa

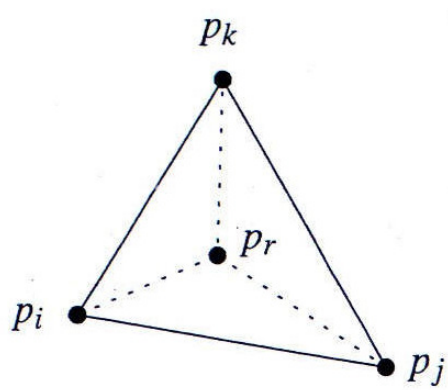
Supuesto: Datos contenidos en triangulación simple



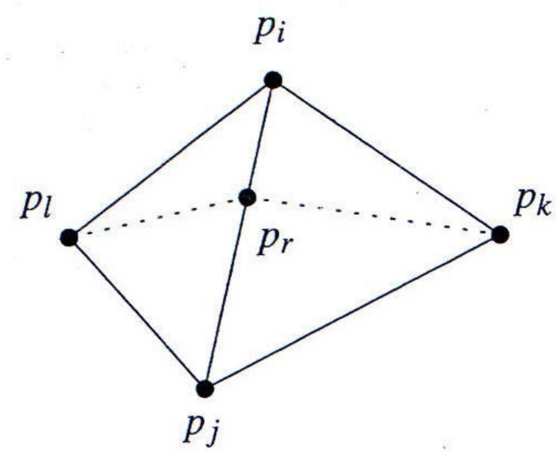
Reduce el número de casos a considerar en el algoritmo

Encontrar triángulo o triángulos que contienen el punto

p_r lies in the interior of a triangle



p_r falls on an edge



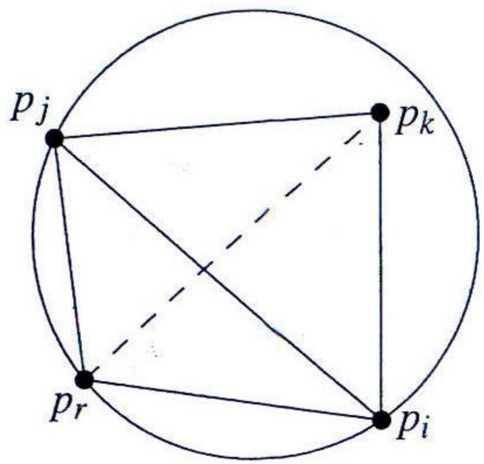
Algoritmo incremental Delaunay

Se insertan los puntos de uno en uno. Elementos fundamentales

- Test-Circulo (t,P) verifica si punto P cumple condición de Delaunay con respecto a triángulo t.
- Operación de intercambio de diagonales
- Supuesto (reduce número de casos): todos puntos contenidos en triangulación inicial simple: triángulo único, rectángulo triangularizado.

Triangulaciones de cuatro puntos

- Dos triangulaciones posibles
- 4 puntos no cocirculares: uno de las triangulaciones es Delauany
- Operación intercambio de diagonales permite delaunizar localmente la triangulación



p_j
p_k
p_r
p_i

Algorithm DELAUNAYTRIANGULATION(P)
Input. A set P of n points in the plane.
Output. A Delaunay triangulation of P.

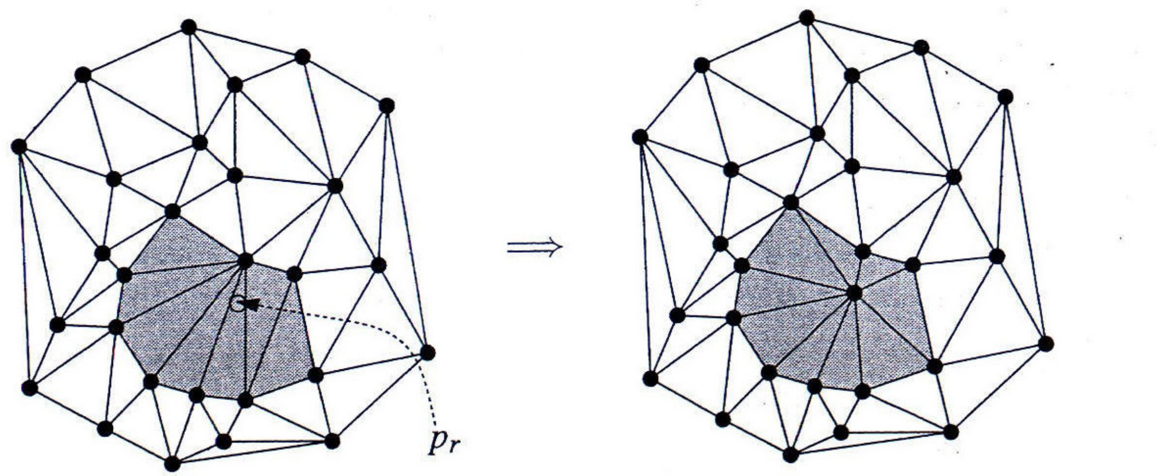
Let p_{-1}, p_{-2}, and p_{-3} be a suitable set of three points such that P is contained in the triangle $p_{-1}p_{-2}p_{-3}$.
Initialize $\mathcal{T}$ as the triangulation consisting of the single triangle $p_{-1}p_{-2}p_{-3}$.
Compute a random permutation $p_1, p_2, \dots, p_n$ of P.
for $r \leftarrow 1$ **to** n
    do (∗ Insert p_r into $\mathcal{T}$: ∗)
        Find a triangle $p_i p_j p_k \in \mathcal{T}$ containing p_r.
        if p_r lies in the interior of the triangle $p_i p_j p_k$
          then Add edges from p_r to the three vertices of $p_i p_j p_k$, thereby splitting $p_i p_j p_k$ into three triangles.
              LEGALIZEEDGE($p_r, \overline{p_i p_j}, \mathcal{T}$)
              LEGALIZEEDGE($p_r, \overline{p_j p_k}, \mathcal{T}$)
              LEGALIZEEDGE($p_r, \overline{p_k p_i}, \mathcal{T}$)
          else (∗ p_r lies on an edge of $p_i p_j p_k$, say the edge $\overline{p_i p_j}$ ∗)
              Add edges from p_r to p_k and to the third vertex p_l of the other triangle that is incident to $\overline{p_i p_j}$, thereby splitting the two triangles incident to $\overline{p_i p_j}$ into four triangles.
              LEGALIZEEDGE($p_r, \overline{p_i p_l}, \mathcal{T}$)
              LEGALIZEEDGE($p_r, \overline{p_l p_j}, \mathcal{T}$)
              LEGALIZEEDGE($p_r, \overline{p_j p_k}, \mathcal{T}$)
              LEGALIZEEDGE($p_r, \overline{p_k p_i}, \mathcal{T}$)
Discard p_{-1}, p_{-2}, and p_{-3} with all their incident edges from $\mathcal{T}$.
return $\mathcal{T}$

Legalize Edge equivale a Test-Circulo

LEGALIZEEDGE($p_r, \overline{p_i p_j}, \mathcal{T}$)

1. (* The point being inserted is p_r, and $\overline{p_i p_j}$ is the edge of $\mathcal{T}$ that may need to be flipped. *)
2. **if** $\overline{p_i p_j}$ is illegal
3.   **then** Let $p_i p_j p_k$ be the triangle adjacent to $p_r p_i p_j$ along $\overline{p_i p_j}$.
4.   (* Flip $\overline{p_i p_j}$: *) Replace $\overline{p_i p_j}$ with $\overline{p_r p_k}$.
5.   LEGALIZEEDGE($p_r, \overline{p_i p_k}, \mathcal{T}$)
6.   LEGALIZEEDGE($p_r, \overline{p_k p_j}, \mathcal{T}$)

Otra implementación: basada en la cavidad



Otras aplicaciones y algoritmos

Triangulación adaptiva para método de elementos finitos: algoritmo de Rivara

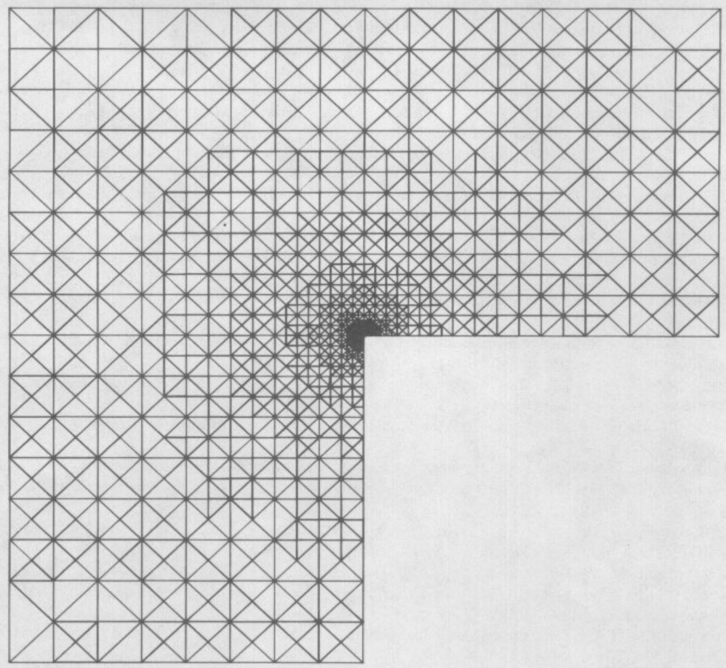


Fig. 13. Last adaptive triangulation obtained for Problem 2.

Refinamiento / Desrefinamiento Rivara

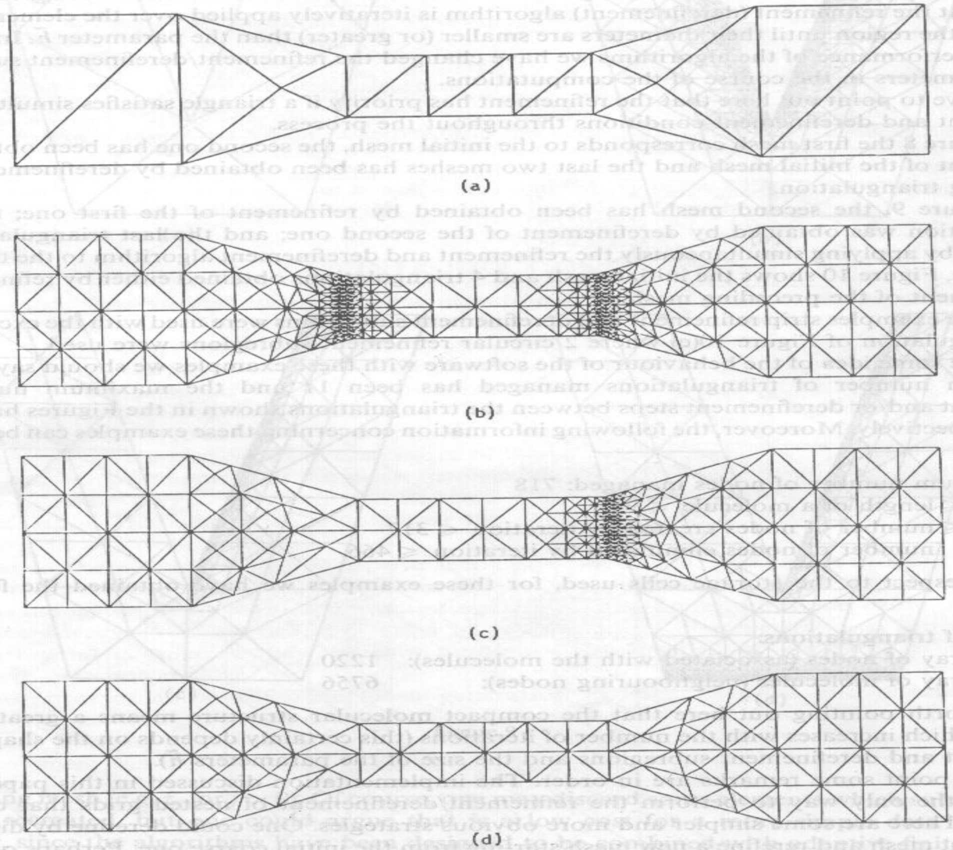
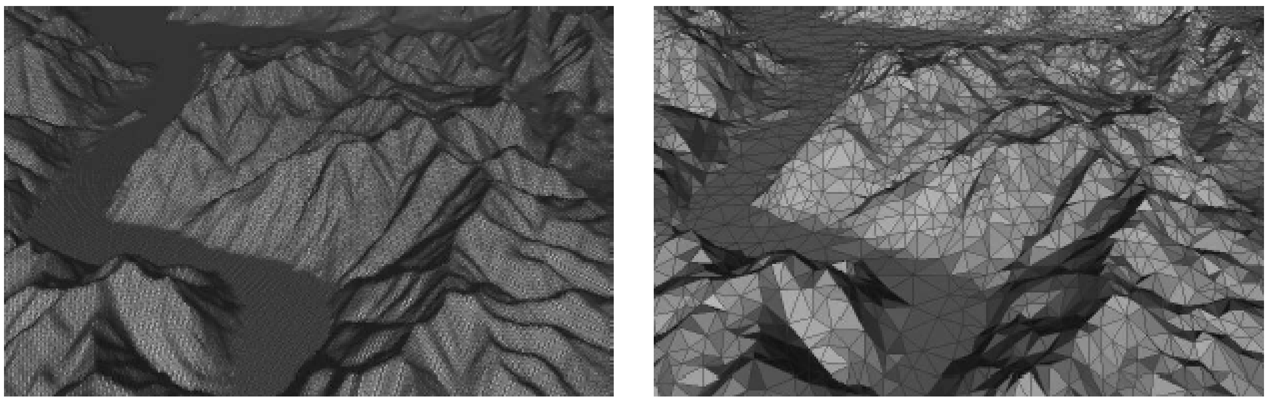


Figure 8

IJNME 97

Simplificación de terrenos: Rivara y otros



(a) (b)

Figure 9: Detail of the Como lake. (a) Original model; (b) Simplified model (5%).

Triangulación de geometrías complejas: comparación Triangle y software DCC (Rivara-Rodriguez)