



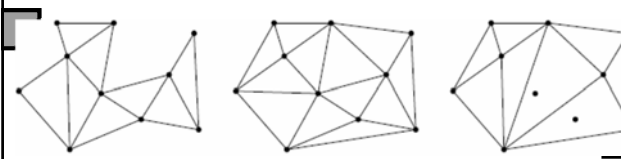
計算幾何学
Computational Geometry



第五章 ドローネ三角形分割
Delaunay Triangulation



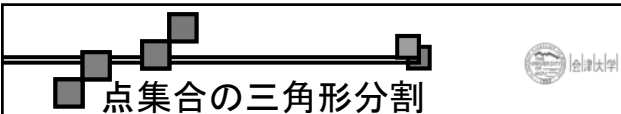
三角形分割



Simple polygon triangulation
単純多角形の
三角形分割

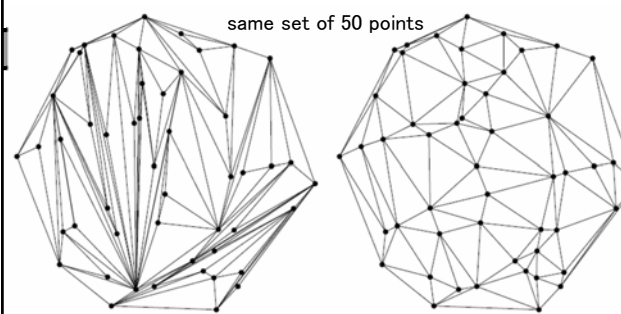
Point set triangulation
点集合の
三角形分割

Not a triangulation
非三角形分割



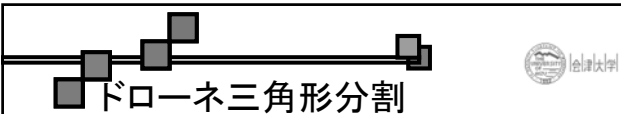
点集合の三角形分割

same set of 50 points

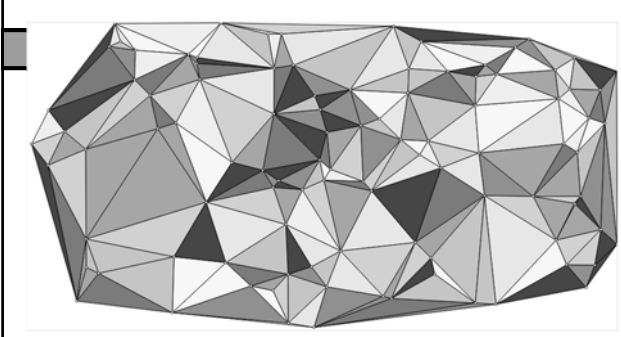
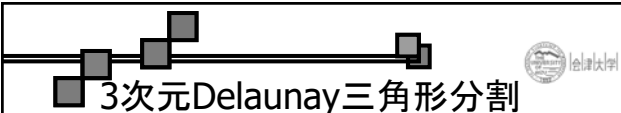


Scan triangulation

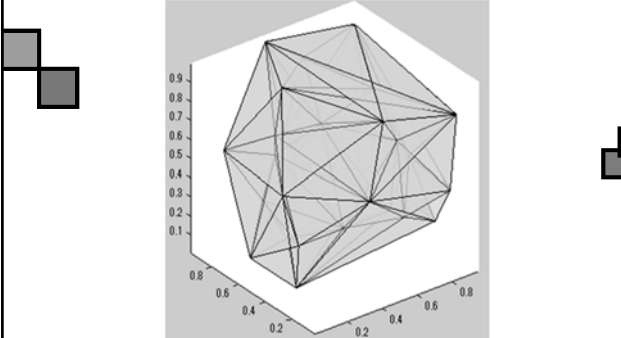
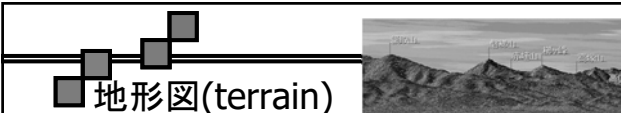
Delaunay triangulation




ドローネ三角形分割

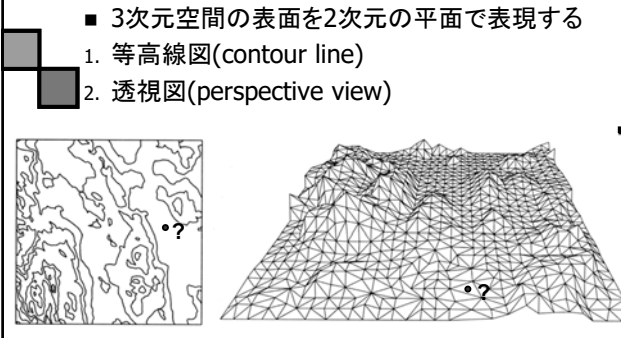
3次元Delaunay三角形分割

地形図(terrain)



- 3次元空間の表面を2次元の平面で表現する
- 1. 等高線図(contour line)
- 2. 透視図(perspective view)



多面体地形図(polyhedral terrain)

- 標本点の高さから他の点の高さを推測する

高さ方向の補間

- 一つの辺を交換するだけで生じた大きな違い
- 最適な方法とは？

問題の提起

- 与えられた平面上の有限点集合に対して、三角形分割の通り数→有限！
- 問題→どの通りの三角形分割が最適？
- 地形図の例を考察してみると→分割角度の大きい方が良さそう？
- 最小角度を最大にすると→最適な三角形分割？

関連術語

- 平面グラフ(planar graph)
頂点の集合は互いに交差しない辺の集合によって結ばれ、平面上に描画したもの
- 連結な平面グラフ(connected planar graph)
平面グラフのどの2頂点間にも辺をたどって行ける
- 平面領域分割(planar subdivision)
平面グラフによって互いに素な領域に分割されること
- 極大平面領域分割(maximal planar subdivision)
平面グラフの結ばれていないどの2頂点を結ぼうとすると必ず既存の辺と交差してしまう際の平面領域分割

平面グラフ(planar graph)

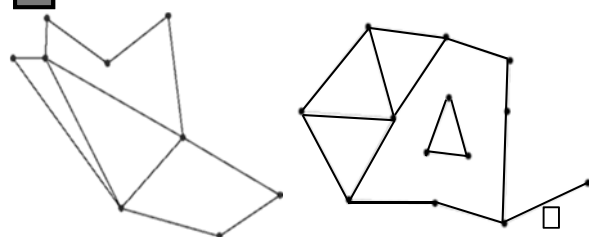
- 頂点の集合は互いに交差しない辺の集合によって結ばれている
- 辺数: planar graph < non-planar graph

連結な平面グラフ(connected planar graph)

- both planar and connected
- planar → without graph edges crossing
- connected → there is a path from any point to any other point in the graph

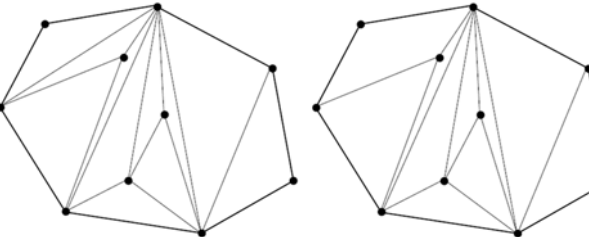
平面領域分割(planar subdivision)

- 平面グラフによって有界でない(非有界)領域を含む互いに素な領域に分割される



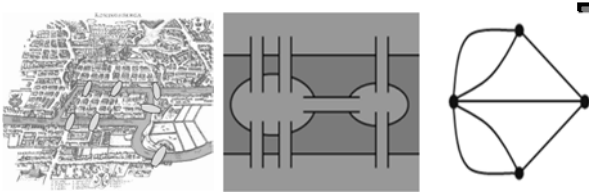
極大平面領域分割(maximal planar subdivision)

- 平面グラフの結ばれていないどの2頂点を結ぼうとすると必ず既存の辺と交差してしまう際の平面領域分割



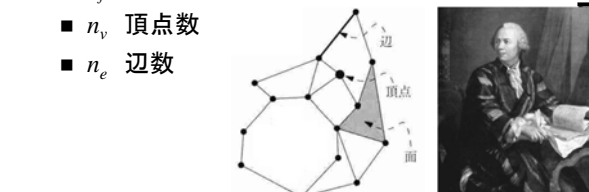
一筆書き(ひとふでがき)

- ケーニヒスベルク問題(Königsberg Bridge Problem)
- 橋を全て1度だけ通って戻ってくるルートが存在する?
→No(1736年、レオンハルト・オイラー) →グラフ論の始まり



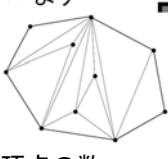
オイラーの公式(Euler's formula)

- 連結な平面グラフに対して、下記の関係式が成り立つ
- n_f 面数(有界な領域+非有界な領域)
- n_v 頂点数
- n_e 辺数




平面点集合の三角形分割

- 平面上の点集合
- P を頂点集合とする極大平面領域分割で、有界な領域がすべて三角形である
- オイラーの公式より
 - 三角形の個数 $= 2n - 2 - k$
 - 辺の数 $= 3n - 3 - k$ n = 頂点の総数
 k = P の凸包の境界上にある頂点の数



三角形の個数と辺の本数

- 総点数 n と凸包境界上の点数 k に依存する
 - Δ の数 $m = 2n - 2 - k$
 - 辺の数 $n_e = 3n - 3 - k$
- 面の数(非有界の面を含む): $n_f = m + 1$ (1)
- 有界の面(Δ)の辺数 $= 3m$
- 非有界の面の辺数 $=$ 凸包境界上の点数 $= k$
- 1辺は2面に共有される→
 実際の辺数: $n_e = (3m + k) / 2$ (2)
- 頂点数: $n_v = n$ (3)
- オイラーの公式: $n_v - n_e + n_f = 2$ (4)
- (1),(2),(3)を(4)に代入: $n - (3m + k) / 2 + m + 1 = 2$
- Δ の数 $m = 2n - 2 - k$ (5)
- 式(5)を(2)に代入, 辺の数 $n_e = 3n - 3 - k$



■ 三角形分割の列挙

- 頂点の総数 = 6
- 凸包頂点数 = 4
- 三角形個数 = 6
- 辺の数 = 11

■ 角度ベクトル(angle-vector)

- 平面上の点集合
- $T=P$ の三角形分割 $\rightarrow m$ 個三角形 $= 3m$ 個内角
- 昇順並び $\alpha_1 < \alpha_2 < \dots < \alpha_{3m}$
- T の角度ベクトル(angle-vector)
 $A(T) = (\alpha_1, \alpha_2, \dots, \alpha_{3m})$, $\alpha_i \geq \alpha_j$ when $i > j$

$A(T) = (15, 15, 30, 45, 45, 80, 80, 90, 150)$

■ 角度最適(angle-optimal)

- 三角形分割 T : $A(T) = (\alpha_1, \alpha_2, \dots, \alpha_{3m})$
- 三角形分割 T' : $A(T') = (\alpha'_1, \alpha'_2, \dots, \alpha'_{3m})$
- $A(T) > A(T') \Leftrightarrow \exists r$ s.t. $\alpha_j = \alpha'_j$, $j = 1, \dots, r$, and $\alpha_{r+1} > \alpha'_{r+1}$
- 全ての T' に成立つ時、 T は角度最適の三角形分割
 \rightarrow How to obtain angle-optimal triangulation?

$A(T') = (15, 15, 25, 40, 40, 65, 75, 125, 150)$ $A(T) = (15, 15, 30, 45, 45, 80, 80, 90, 150)$

■ 辺のフリップ(edge flip)

- 凸四角形 $p_i p_j p_k p_l$
- 線分 $p_i p_k$ による三角形分割 T
- 線分 $p_j p_l$ による三角形分割 T'
- 辺のフリップ $\rightarrow T \rightarrow T' \rightarrow$ 角度ベクトル $A(T) \Rightarrow A(T')$

■ 不正な辺(illegal edge)

- $A(T) = (\alpha_1, \alpha_2, \dots, \alpha_6)$
- $A(T') = (\alpha'_1, \alpha'_2, \dots, \alpha'_6)$
- もし $\alpha_1 < \alpha'_1$ (最小角度だけを比べる)
 \rightarrow =不正な辺 \rightarrow 判断方法?

■ タレスの定理

- Thales' Theorem can be used to test if an edge is legal without calculating angles

Let C be a circle, l a line intersecting C in points a and b and p, q, r , and s points lying on the same side of l .

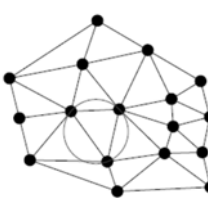
Suppose that p and q lie on C , that r lies inside C , and that s lies outside C .

Then: $\angle arb > \angle apb = \angle aqb > \angle asb$.

Empty Circle Property

Theorem

- Let P be a set of points in the plane, and let T be a triangulation of P . Then T is a Delaunay triangulation of P if and only if the circumcircle of any triangle of T does not contain a point of P in its interior.



Angle Optimality and Delaunay Triangulations

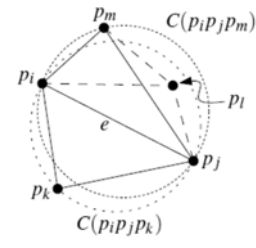
Theorem

- Let P be a set of points in the plane. Any angle-optimal triangulation of P is a Delaunay triangulation of P .
- Furthermore, any Delaunay triangulation of P maximizes the minimum angle over all triangulations of P .

Legal Triangulations and Delaunay Triangulations

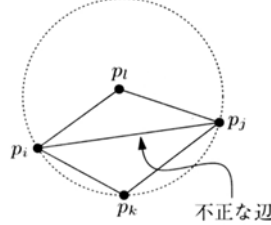
Theorem

- Let P be a set of points in the plane. A triangulation T of P is legal if and only if T is a Delaunay triangulation.



不正な辺の判断方法

■ 辺 $p_i p_j$ が不正である必要十分条件:
 p_l が p_i, p_j, p_k を通る円の内部に含まれる



不正な辺

ドローネ三角形分割の求め方

- 不当な辺フリップ法
 - 力任せ法 (Brute force method)
- 逐次添加法
- ボロノイ図平面双対法 ($\text{Vor}(P) \rightarrow \mathcal{Dg}(P)$)

不正な辺フリップ法 (正当な三角形分割) Illegal Edge Flip Method (Legal Triangulation)

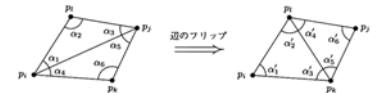
基本的な考え方

- 不正な辺を含まない三角形分割
- 角度最適な三角形分割
- 求め方:
 1. 任意の初期三角形分割から出発
 2. 不正な辺があれば、辺のフリップを行う
 3. すべての辺が正当になるまで、繰り返す

アルゴリズム

LegalTriangulation(\mathcal{T})

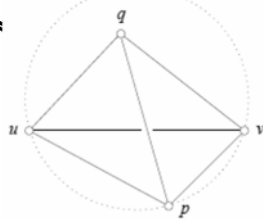
- 入力: 点集合 P のある任意の三角形分割 \mathcal{T}
 - 出力: P の正当な三角形分割
1. while (\mathcal{T} が不正な辺 を含んでいる) do
 2. EdgeFlipping()
 3. \mathcal{T} から を取り除く
 4. \mathcal{T} に を加える
 5. return \mathcal{T}



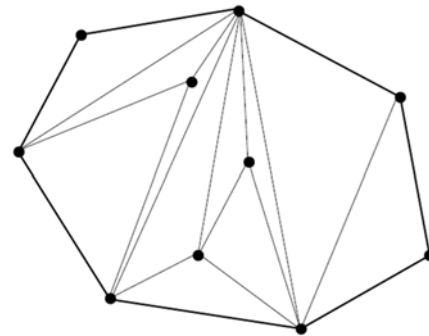
アルゴリズム

EdgeFlipping(uv)

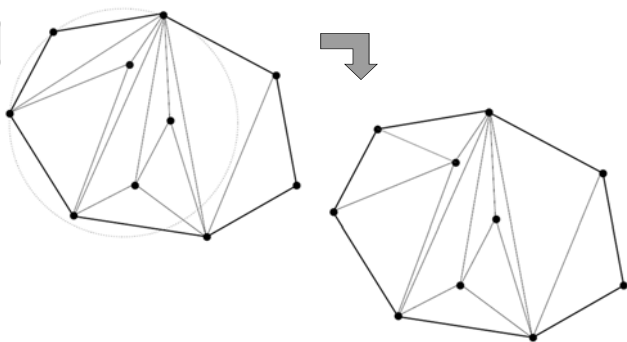
1. unmark edge uv
2. if uv is illegal then
3. substitute pq for uv
4. for $ab \in \{up, pv, vq, qu\}$ do
5. if ab is unmarked then
6. push ab on the stack
7. and mark it
8. endif
9. endfor
10. endif



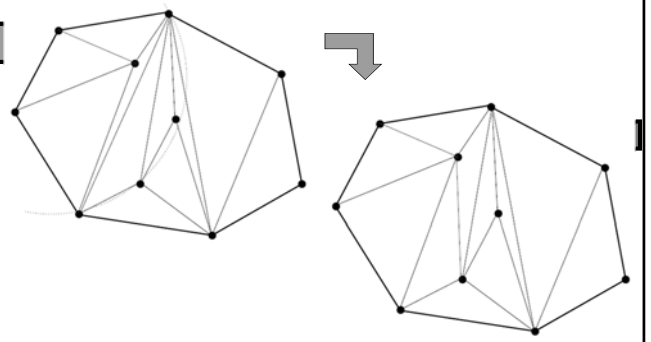
Legal Triangulation – initial

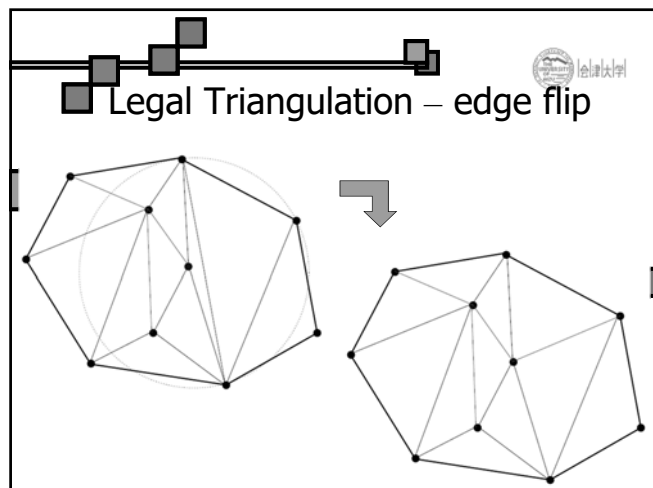
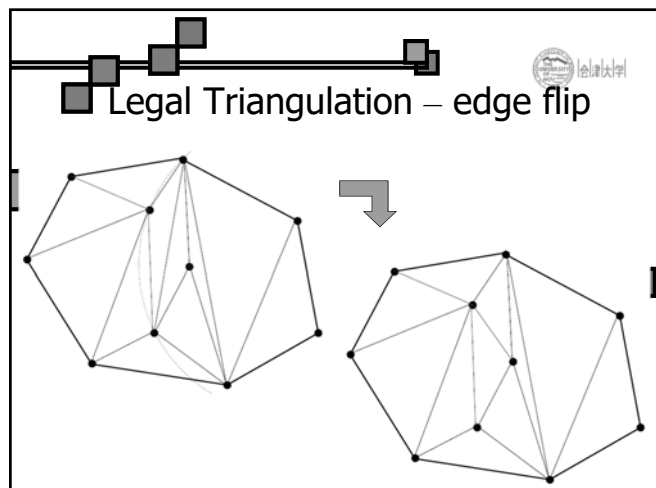


Legal Triangulation – edge flip



Legal Triangulation – edge flip





計算量

- n 点集合
- 線分の組合せ数 $\binom{n}{2} = n(n-1)/2$
- EdgeFlipping処理回数 = 線分の数
- 全部の計算量 = $O(n^2)$

逐次添加法
Random Incremental Method

基本的な考え方

1. 点集合 P を含む大きな三角形 $\Delta p_{-1}p_{-2}p_{-3}$ から始める
2. ランダムな順序で任意の点 p_r を加えて、現在の三角形分割において p_r の入っている三角形の領域又は辺を求める

基本的な考え方 続き

3. 領域 $\Rightarrow p_r$ からこの三角形の三つの頂点へそれぞれ辺を加え、新たな三角形分割を構成する
4. 辺 $\Rightarrow p_r$ からこの辺を共有する2つの三角形の対角頂点へそれぞれ辺を加え、新たな三角形分割を構成する
5. 辺の正当性を検証し、不正な辺をフリップする

■ アルゴリズム

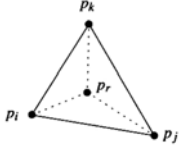
DelaunayTriangulation(P)

- 入力: 平面上の n 個の点集合 P
- 出力: P のドローネ三角形分割 T

1. P が $\Delta p_{-1}p_{-2}p_{-3}$ に含まれるように点 p_{-1}, p_{-2}, p_{-3} を選ぶ
2. $\Delta p_{-1}p_{-2}p_{-3}$ を最初の三角形分割 T とする
3. for $r=1$ to n {
4. do (* p_r を T に挿入する *)
5. p_r の入っている $\Delta p_i p_j p_k \in T$ を求める
6. if p_r が $\Delta p_i p_j p_k$ の内部にある

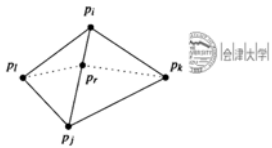
■ アルゴリズム 続き

7. then
8. p_r と $\Delta p_i p_j p_k$ の3頂点を辺で結んで、新たな三角形分割を構成する
9. LegalizeEdge(p_r, p_i, T)
10. LegalizeEdge(p_r, p_j, T)
11. LegalizeEdge(p_r, p_k, T)



■ アルゴリズム 続き

12. else
13. p_r と四角形 $p_i p_j p_k p_l$ の対角頂点を辺で結んで、新たな三角形分割を構成する
14. LegalizeEdge(p_r, p_i, T)
15. LegalizeEdge(p_r, p_j, T)
16. LegalizeEdge(p_r, p_k, T)
17. LegalizeEdge(p_r, p_l, T)
18. } // end of for
19. 点 p_{-1}, p_{-2}, p_{-3} と、これらに接続する辺を T から取り除く
20. return T

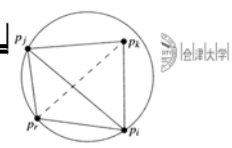


■ アルゴリズム

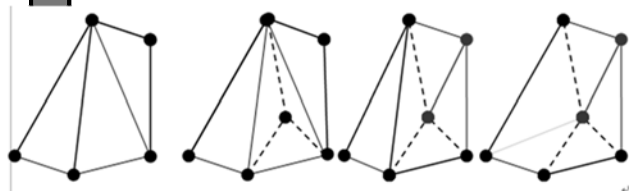
LegalizeEdge(p_r, p_i, T)

- 入力: $p_r \Rightarrow$ 挿入頂点, $T \Rightarrow$ 今までの三角形分割
- ⇒ 正当性検証の辺
- 出力: 新たな三角形分割 T

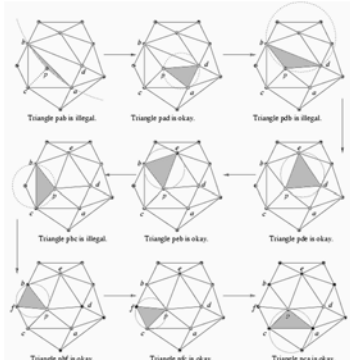
1. if p_r が $\Delta p_i p_j p_k$ の外部にある ⇒ 不正な辺
2. then
3. $\Delta p_i p_j p_k$ をフリップし、 $\Delta p_i p_r p_k$ で置き換える
4. LegalizeEdge(p_r, p_j, T)
5. LegalizeEdge(p_r, p_k, T)

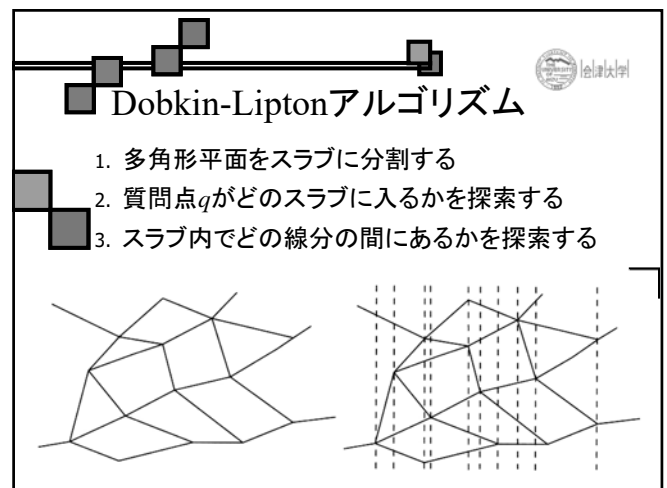
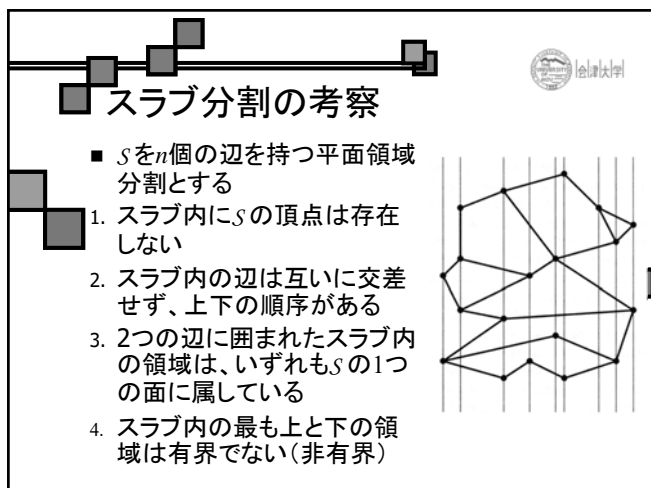
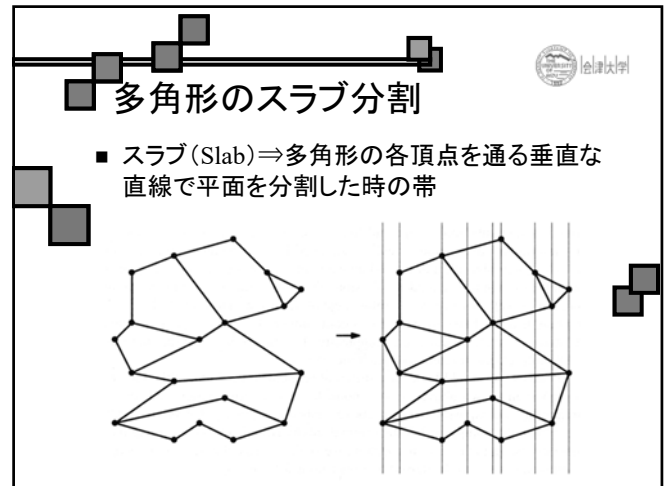
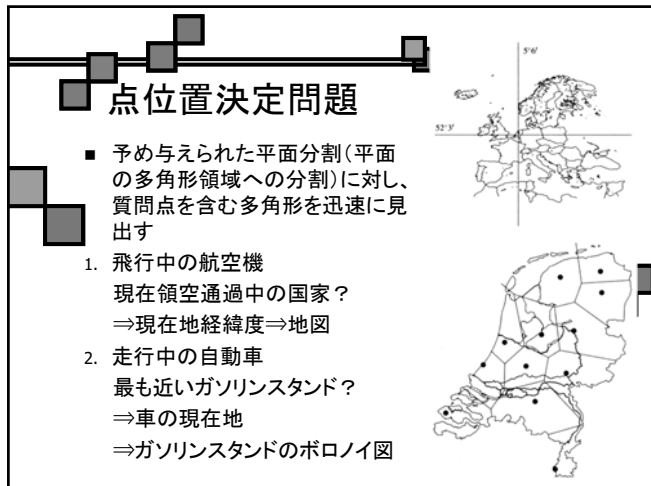
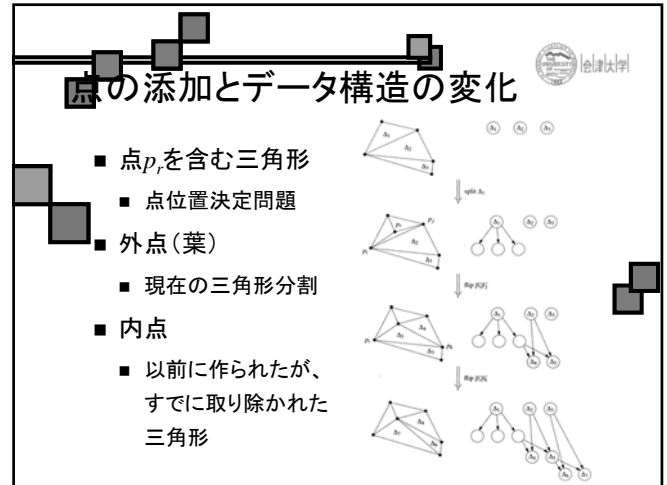
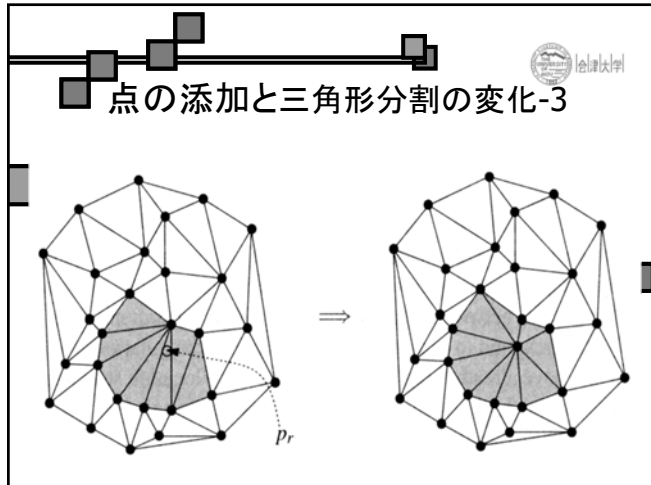


■ 点の添加と三角形分割の変化-1



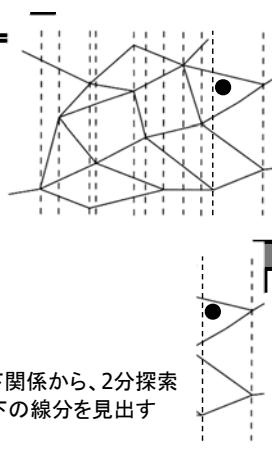
■ 点の添加と三角形分割の変化-2





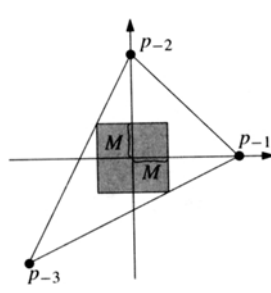
■ アルゴリズム

1. 平面領域分割の頂点のx座標値を蓄えた配列を生成
2. 質問点 q のx座標値を用いて、上記の配列に関する2分探索 $\Rightarrow q$ を含むスラブを見出す
3. 見つけたスラブ内の辺の上下関係から、2分探索を行い、 q のある面の直ぐ上下の線分を見出す



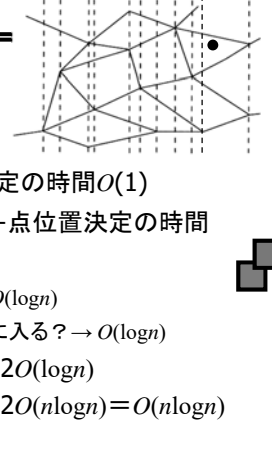
■ 点 p_{-1}, p_{-2}, p_{-3} の選び方

1. $M \leftarrow$ 点集 P の点のx又はy座標値の絶対値の最大値
2. $p_{-1} = (3M, 0)$
3. $p_{-2} = (0, 3M)$
4. $p_{-3} = (-3M, -3M)$



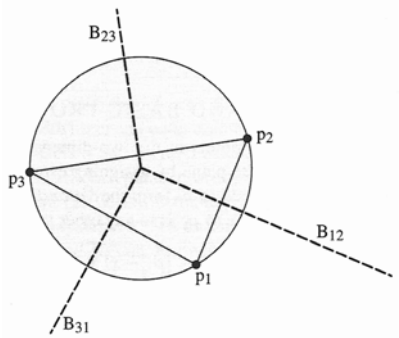
■ 計算量

- 1つ三角形の生成は一定の時間 $O(1)$
- 全部の計算量 $\rightarrow O(n) + \text{点位置決定の時間}$
- 点位置決定問題
 - どのスラブに入る? $\rightarrow O(\log n)$
 - スラブ内のどの線分間に入る? $\rightarrow O(\log n)$
- 一点の計算量 $= O(1) + 2O(\log n)$
- 全部の計算量 $= O(n) + 2O(n \log n) = O(n \log n)$



■ ボロノイ図平面双対法 Dual Graph of Voronoi ($\text{Vor}(P) \rightarrow \text{Dg}(P)$)

■ 外接円と垂直二等分線



■ 双対変換(duality transform)

- 基本概念:

双対変換を用いると、平面上の点集合に関する色々な性質を表すことができる。また点集合を眺めていても、上手く解けないような問題が双対変換によって簡単に解けることがよくある。

双対変換(duality transform)

■ 基本性質:

主平面(primal plane) \Leftrightarrow 双対平面(dual plane)

1. 接続関係不変 (incidence preserving)
1対1の映写
2. 順序関係不変 (order preserving)
上下位置関係変わらず (主平面と双対平面における点と線の位置関係)

双対変換(duality transform)

■ 具体の例:

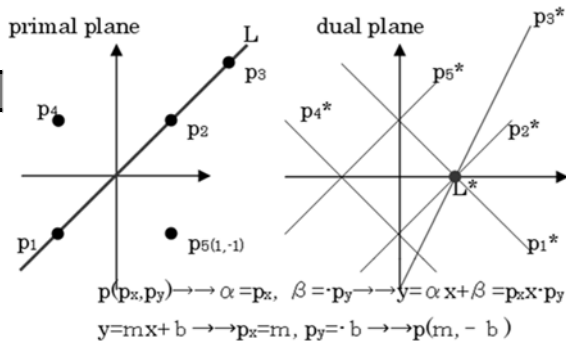
- 平面上1点 $p(x, y) \rightarrow 2$ パラメータ (座標 x, y)
- 平面上1直線 $y = \alpha x + \beta \rightarrow 2$ パラメータ (傾き α , 切片 β)
- We can map a set of points to a set of lines, and vice versa, in a one-to-one manner.
- 色々な映写方法がある
 - 直線 \Leftrightarrow 点
 - 1直線とその上の3点 \Leftrightarrow 同じ点を通る3本直線

双対変換(duality transform)

■ 方法:

1. 点 \rightarrow 直線
 - 点 $p := (p_x, p_y) \rightarrow$ 直線 $p^*: y = p_x x - p_y$
2. 直線 \rightarrow 点
 - 直線 $L: y = mx + b \rightarrow$ 点 $L^* := (m, -b)$

双対変換(duality transform)



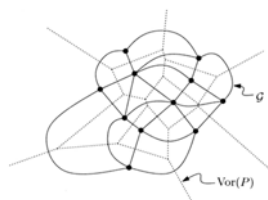
ボロノイ図の双対グラフ

ボロノイ図 $\text{Vor}(P)$

- 母点
- 母点間のボロノイ辺
- 頂点

グラフ g

- 頂点
- 頂点間の辺(枝)
- 有界の面

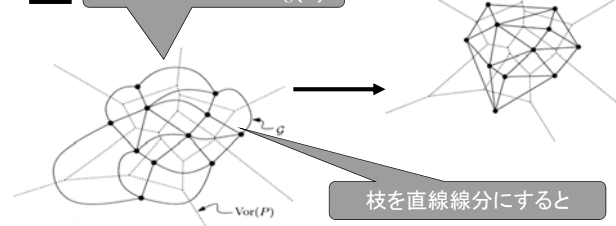


ドローネグラフ(Delaunay graph)

Boris Nikolaevich Delone
1890–1980, ロシア数学者

P のドローネ三角形分割

P のドローネグラフ $Dg(P)$



ドローネ辺の判断方法

円心はボロノイ領域 $\mathcal{V}(p_i)$ と $\mathcal{V}(p_j)$ の共有辺にあって、母点 p_i, p_j を通る円内に他の母点がないような円が存在する

まとめ

- P を平面上の点集合とし、 T を P の三角形分割とする
- 1. 3点 $p_i, p_j, p_k \in P$ がドローネグラフの同じ面の頂点であるための必要十分条件は、 p_i, p_j, p_k を通る円の内部に P の点を含まないこと
- 2. 2点 $p_i, p_j \in P$ の線分 $p_i p_j$ が P のドローネグラフの辺であるための必要十分条件は、 P の他の点を含まない、 p_i, p_j を通る円が存在すること
- 3. T が P のドローネ三角形分割であるための必要十分条件は、 T の任意1つ三角形の外接円の内部に P の点を含まないこと
- 4. P の三角形分割 T が正当であるための必要十分条件は、 T が P のドローネ三角形分割であること
- 5. P の角度最適な三角形分割 $\Leftrightarrow P$ のドローネ三角形分割 \Leftrightarrow 最小角度を最大にする

応用

- Geographical Information System
 - Terrain Height Interpolation
- Computer Graphics
 - Triangular Mesh Generation
 - Surface Reconstruction
- Computer Simulation
 - FEM Modeling

Geographical Information System

Computer Graphics

Computer Simulation