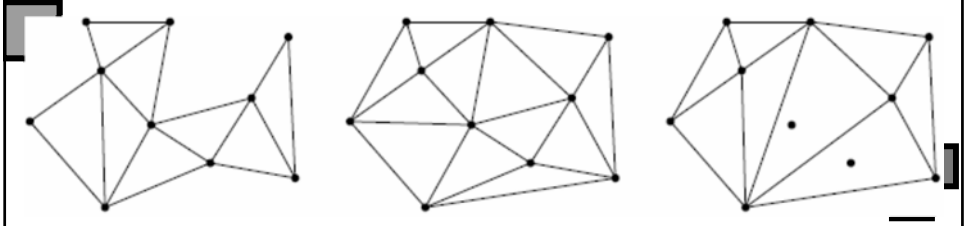
計算幾何学
Computational Geometry

会津大学

第五章 ドローネ三角形分割
Delaunay Triangulation

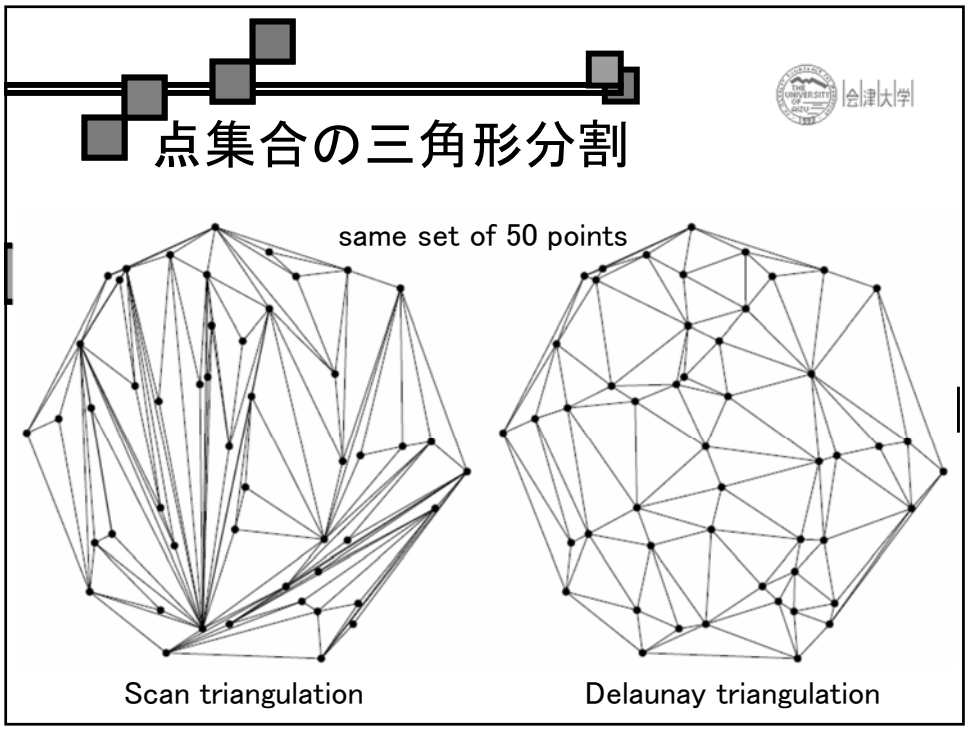
三角形分割

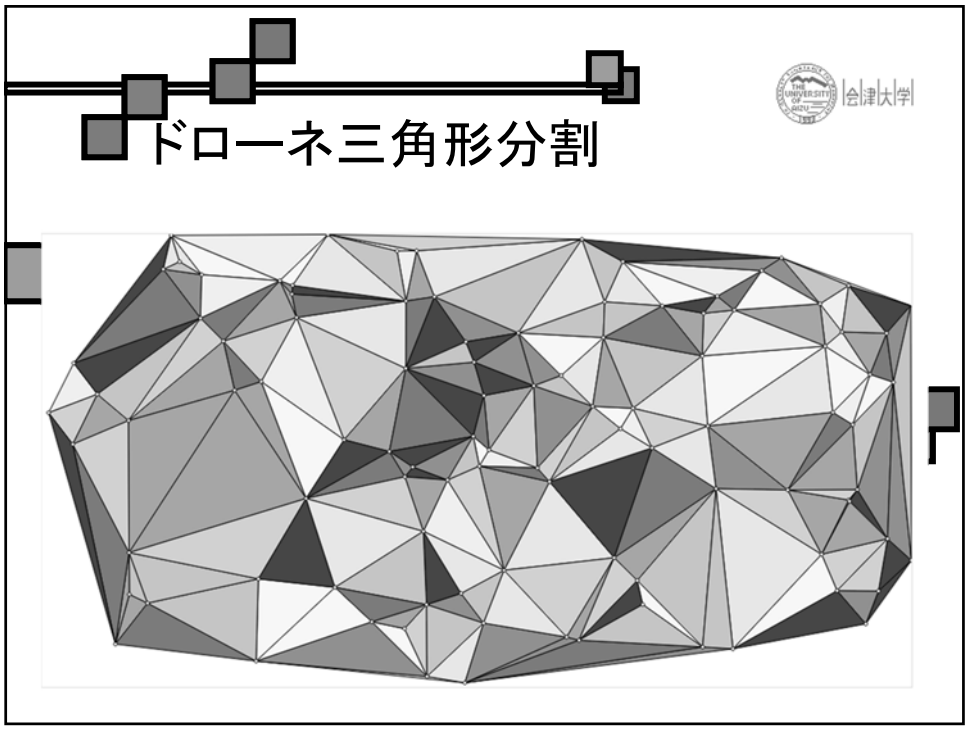


Simple polygon triangulation
単純多角形の三角形分割

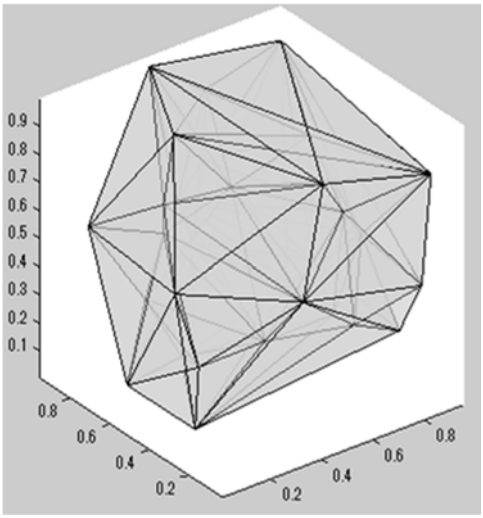
Point set triangulation
点集合の三角形分割

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

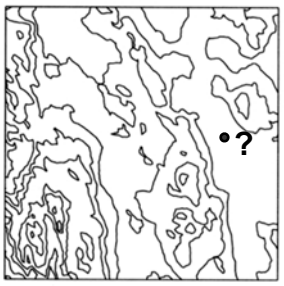
3次元Delaunay三角形分割

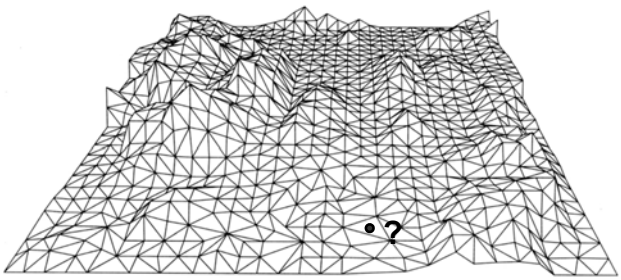


地形図(terrain)

- 3次元空間の表面を2次元の平面で表現する

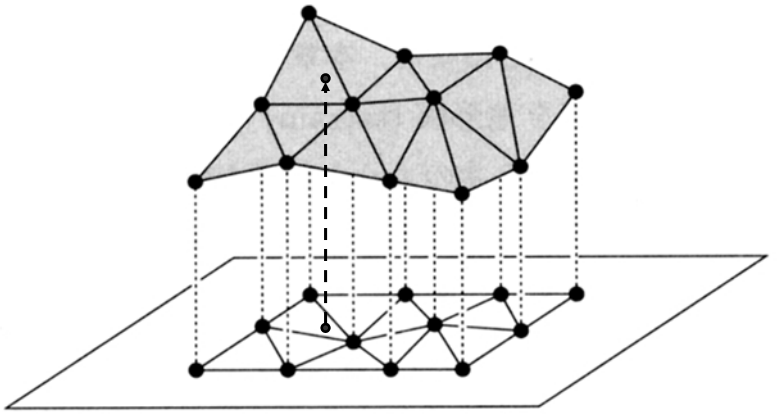
1. 等高線図(contour line)
2. 透視図(perspective view)



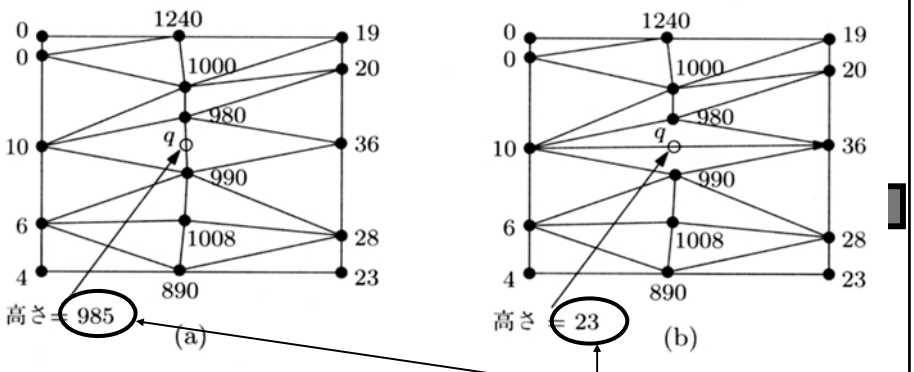


多面体地形図(polyhedral terrain)

- 標本点の高さから他の点の高さを推測する



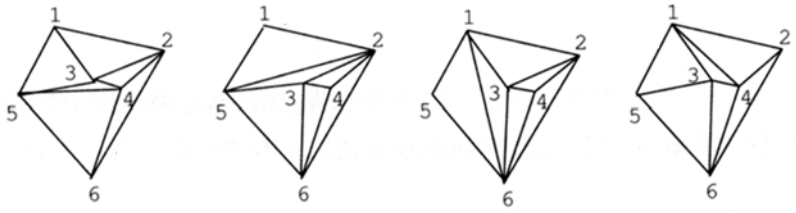
高さ方向の補間



- 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？

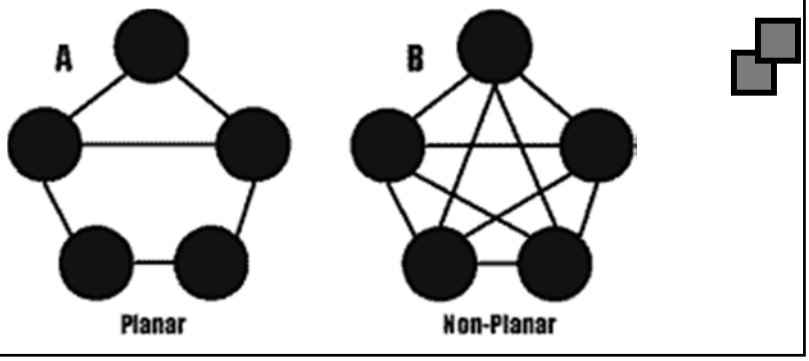


関連術語

- 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

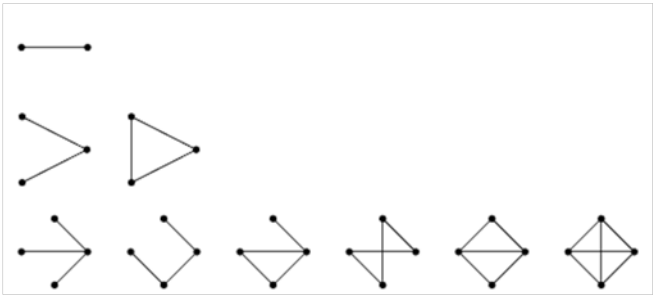
平面グラフ(planar graph)

- 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



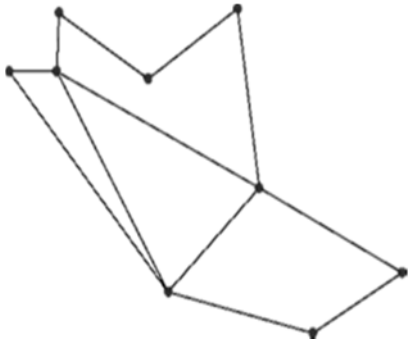
連結な平面グラフ(connected planar graph)

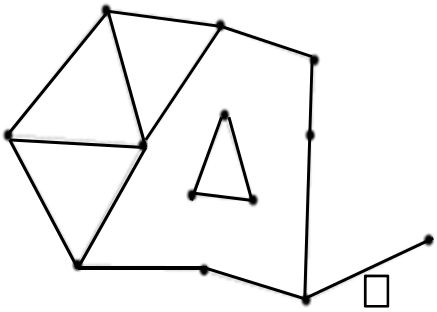
- both planar and connected
- planar →without graph edges crossing
- connected →there is a path from any point to any other point in the graph



平面領域分割(planar subdivision)

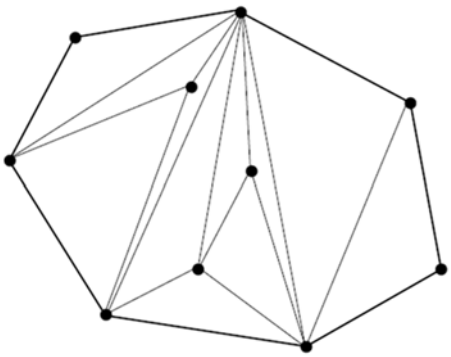
- 平面グラフによって有界でない（非有界）領域を含む互いに素な領域に分割される

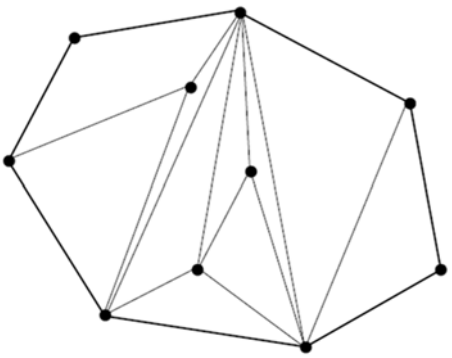




極大平面領域分割(maximal planar subdivision)

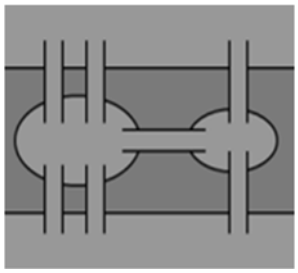
- 平面グラフの結ばれていないどの2頂点を結ぼうとすると必ず既存の辺と交差してしまう際の平面領域分割





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

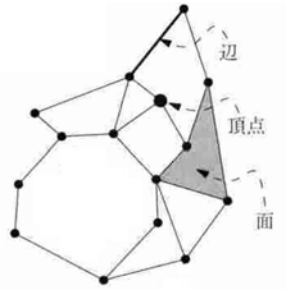




オイラーの公式(Euler's formula)

- 連結な平面グラフに対して、下記の関係式が成り立つ

$$n_v - n_e + n_f = 2$$

- n_f 面数（有界な領域＋非有界な領域）
- n_v 頂点数
- n_e 辺数



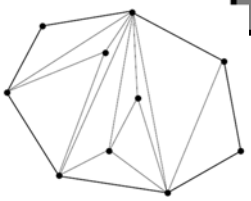
平面点集合の三角形分割

- 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

1. 三角形の個数 = $2n-2-k$
2. 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

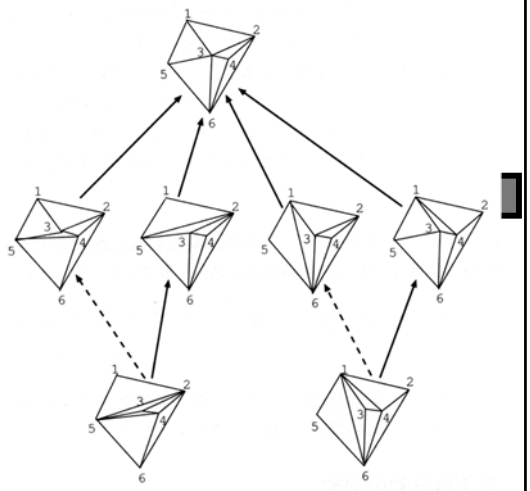


三角形の個数と辺の本数

- 総点数nと凸包境界上の点数kに依存する
  - ⊿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(⊿)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: $n-(3m+k)/2+m+1=2$
- ⊿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

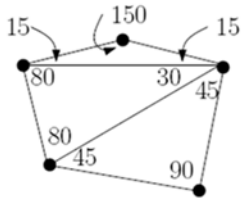
- 頂点の総数 =6
- 凸包頂点数 =4
- 三角形個数 =6
- 辺の数 =11



角度ベクトル(angle-vector)

- 平面上の点集合
- T＝Pの三角形分割→m個三角形＝3m個内角
- 昇順並び$\alpha_1<\alpha_2<\ldots<\alpha_{3m}$
- Tの角度ベクトル(angle-vector)

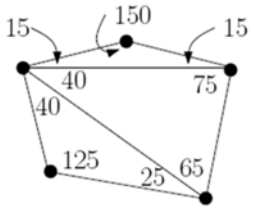
  $A(T)=(\alpha_1,\alpha_2,\ldots,\alpha_{3m})$, $\alpha_i \geq \alpha_j$ when $i>j$



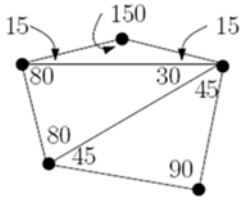
$A(T) = (15, 15, 30, 45, 45, 80, 80, 90, 150)$

角度最適(angle-optimal)

- 三角形分割T: $A(T)=(\alpha_1,\alpha_2,\ldots,\alpha_{3m})$
- 三角形分割T': $A(T')=(\alpha'_1,\alpha'_2,\ldots,\alpha'_{3m})$
- $A(T)>A(T')\Leftrightarrow\exists r$ s.t. $\alpha_j=\alpha'_j$, $j=1,\ldots r$, 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
  →How to obtain angle-optimal triangulation?



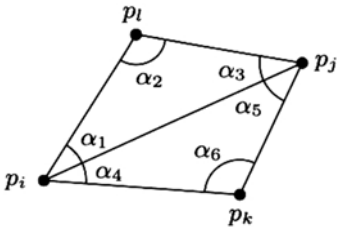
$A(T') = (15, 15, 25, 40, 40, 65, 75, 125, 150)$



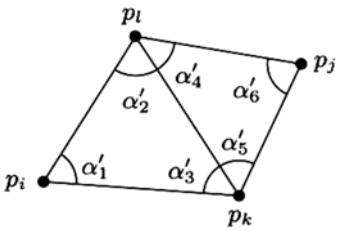
$A(T) = (15, 15, 30, 45, 45, 80, 80, 90, 150)$

辺のフリップ(edge flip)

- 凸四角形$p_ip_lp_jp_k$
- 線分 p_lp_j による三角形分割T
- 線分 p_ip_k による三角形分割T'
- 辺のフリップ→$T\Rightarrow T'$→角度ベクトルA(T)⇒A(T')



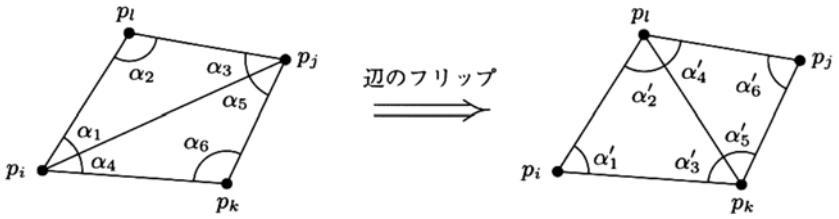
辺のフリップ



不正な辺(illegal edge)

- A(T)=(a_1,a_2,…,a_6)
- A(T')=(a_1',a_2',…,a_6')
- もし (最小角度だけを比べる)

→ =不正な辺→ 判断方法？

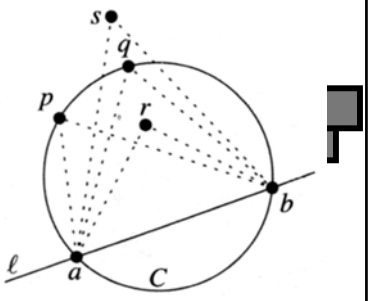


タレスの定理

- *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.

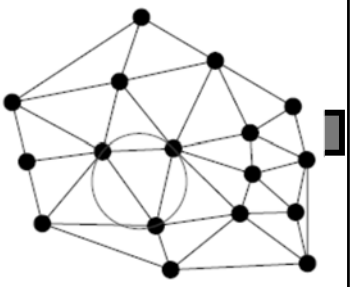


Then: $\angle arb > \angle apb = \angle aqb > \angle asb.$

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

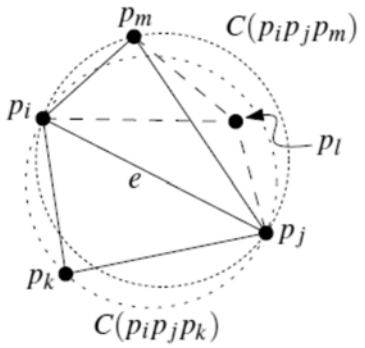
Theorem

- 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

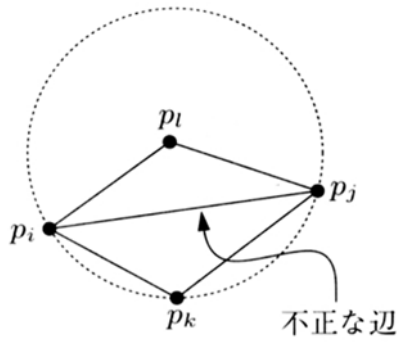
- Let P be a set of points in the plane. A triangulation T of P is legal if and only if T is a Delaunay triangulation.



不正な辺の判断方法

- 辺 p_ip_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



ドローネ三角形分割の求め方

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

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

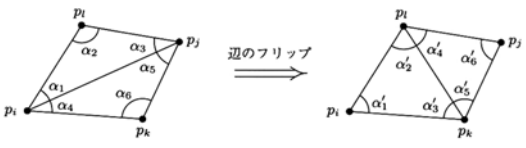
- 不正な辺を含まない三角形分割
- 角度最適な三角形分割
- 求め方:
  1. 任意の初期三角形分割から出発
  2. 不正な辺があれば、辺のフリップを行う
  3. すべての辺が正当になるまで、繰り返す

アルゴリズム

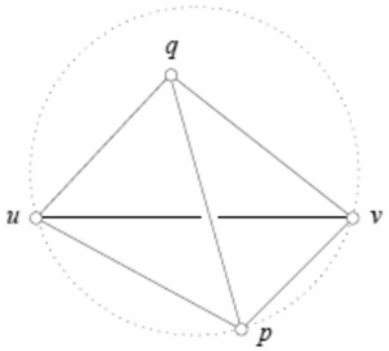
LegalTriangulation($\mathcal{T}$)

- 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 $p_i p_j$ を含んでいる) do
2. EdgeFlipping($p_i p_j$)
3. $\mathcal{T}$から $p_i p_j$ を取り除く
4. $\mathcal{T}$に $p_k p_l$ を加える
5. return $\mathcal{T}$



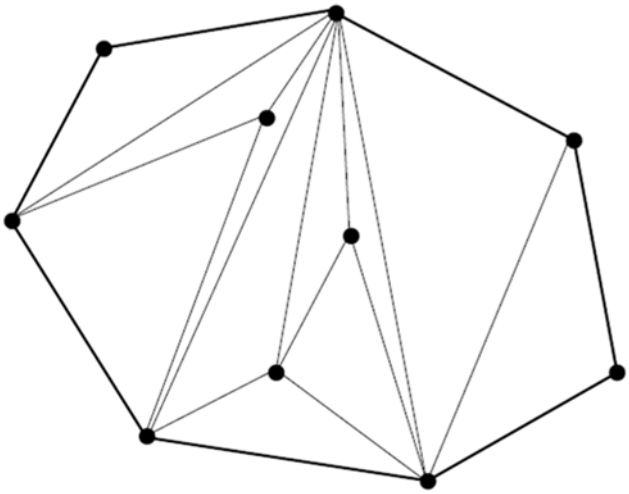
アルゴリズム

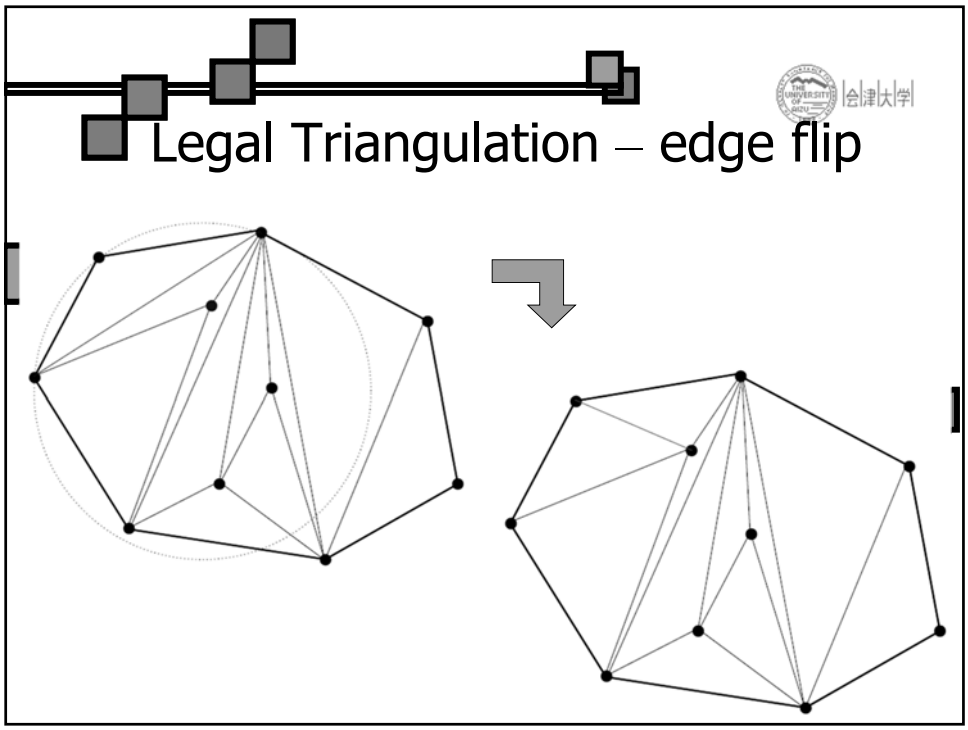


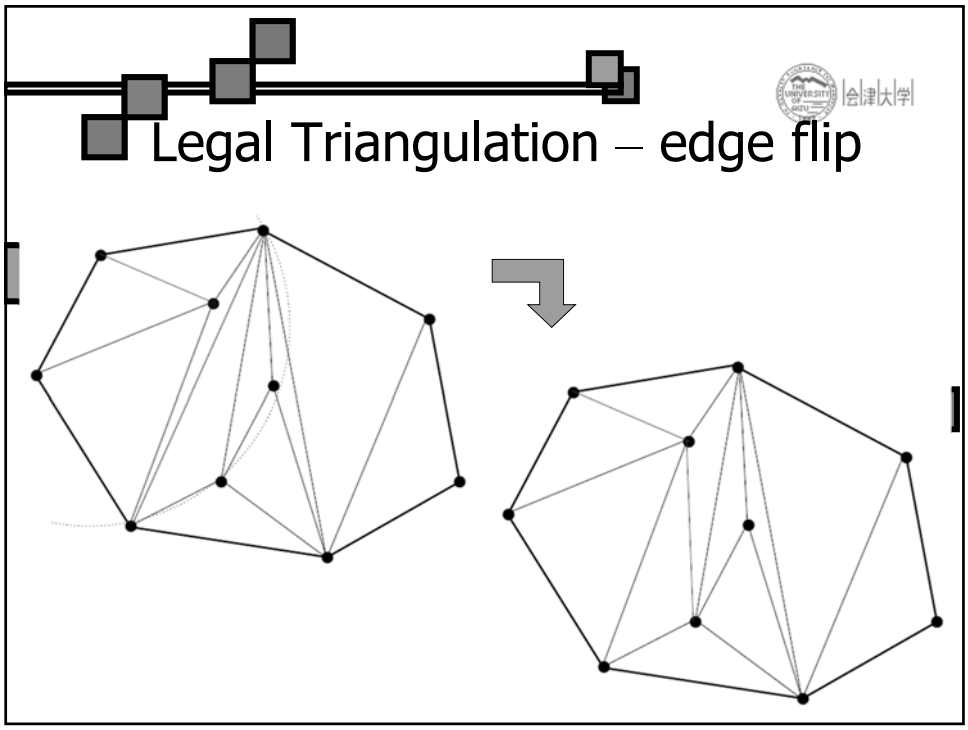
EdgeFlipping (*uv*)

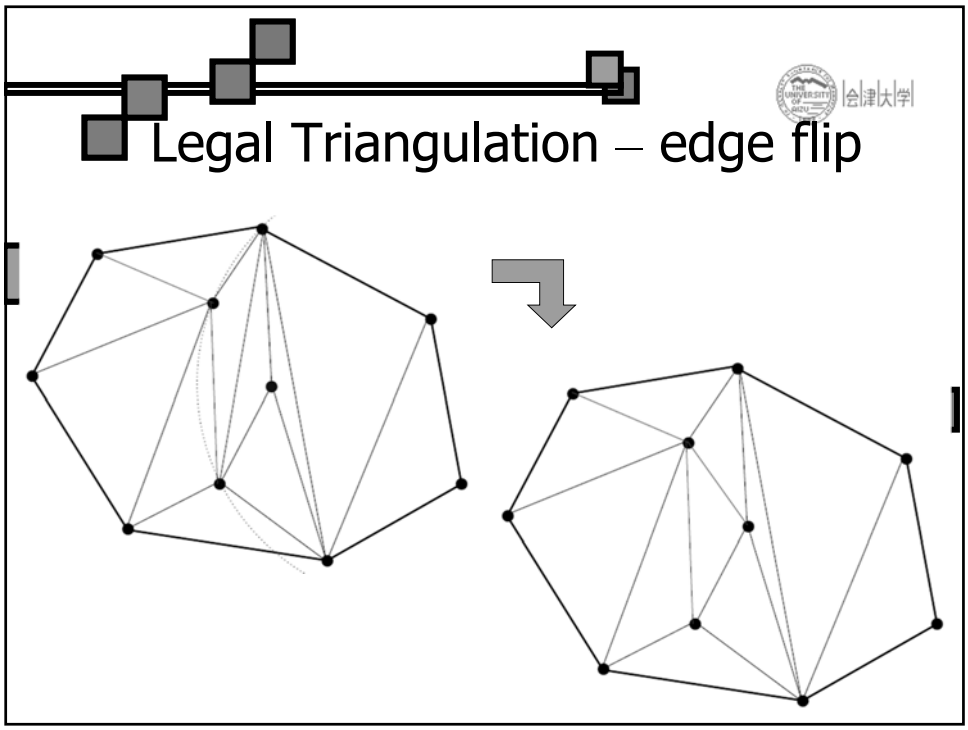
1. unmark edge *uv*
2. if *uv* is illegal then
3. substitute *pq* for *uv*
4. for *ab* ∈ {*up*, *pv*, *vq*, *qu*} do
5. if *ab* is unmarked then
6. push *ab* on the stack
7. and mark it
8. endif
9. endfor
10. endif

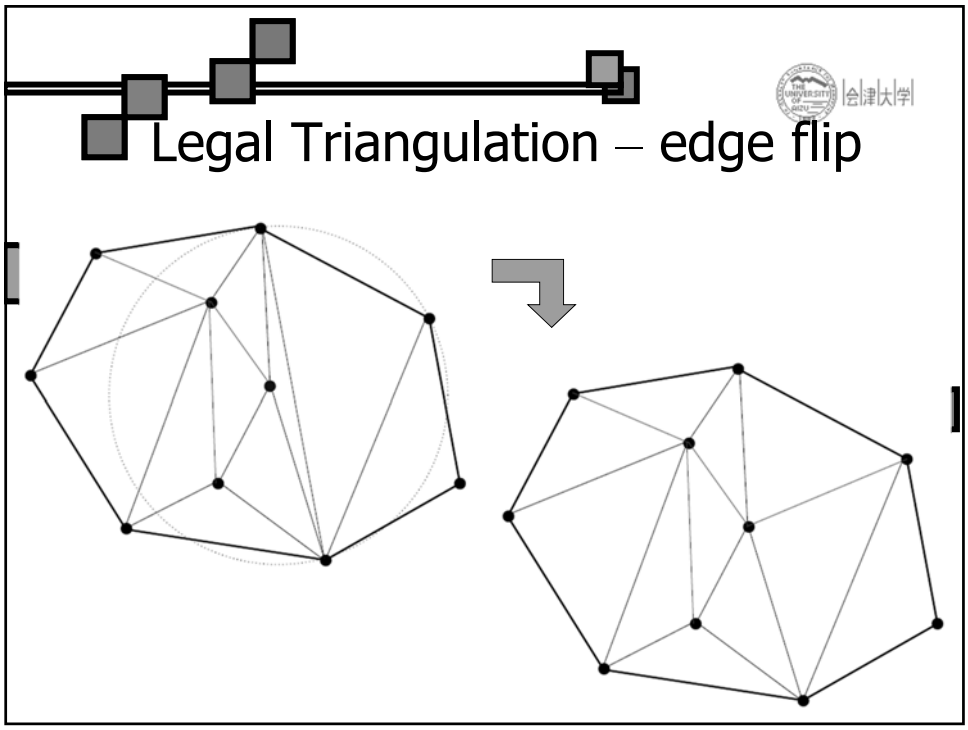
Legal Triangulation – initial




Legal Triangulation – edge flip


Legal Triangulation – edge flip


Legal Triangulation – edge flip


Legal Triangulation – edge flip

計算量

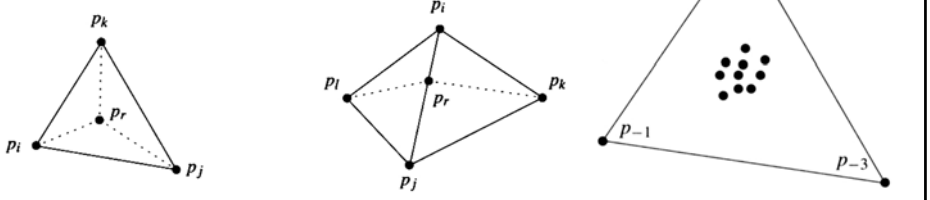
- n点集合
- 線分の組合せ数 $\binom{n}{2} = n(n-1)/2$
- EdgeFlipping処理回数＝線分の数
- 全部の計算量＝$O(n^2)$

逐次添加法

Random Incremental Method

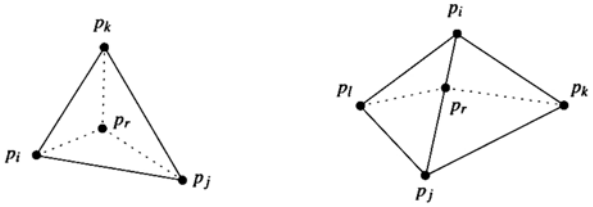
基本的な考え方

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⇒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⇒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

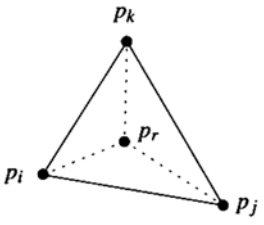
DelaunayTriangulation(P)

- 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$\mathcal{T}$

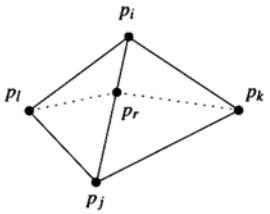
1. Pが$\triangle p_{-1}p_{-2}p_{-3}$に含まれるように点$p_{-1}$, p_{-2}, p_{-3}を選ぶ
2. $\triangle p_{-1}p_{-2}p_{-3}$を最初の三角形分割$\mathcal{T}$とする
3. for r=1 to n {
4. do (* p_rを$\mathcal{T}$ に挿入する *)
5. p_rの入っている$\triangle p_ip_jp_k \in \mathcal{T}$ を求める
6. if p_rが$\triangle p_ip_jp_k$の内部にある

アルゴリズム続き

7. then
8. p_rと$\triangle p_ip_jp_k$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, , $\mathcal{T}$)
10. LegalizeEdge(p_r, , $\mathcal{T}$)
11. LegalizeEdge(p_r, , $\mathcal{T}$)

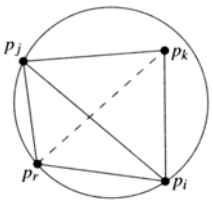


アルゴリズム続き



12. else
13. p_rと四角形$p_ip_jp_kp_l$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, , $\mathcal{T}$)
15. LegalizeEdge(p_r, , $\mathcal{T}$)
16. LegalizeEdge(p_r, , $\mathcal{T}$)
17. LegalizeEdge(p_r, , $\mathcal{T}$)
18. } // end of for
19. 点p_{-1}, p_{-2}, p_{-3}と、これらに接続する辺を$\mathcal{T}$ から取り除く
20. return $\mathcal{T}$

アルゴリズム

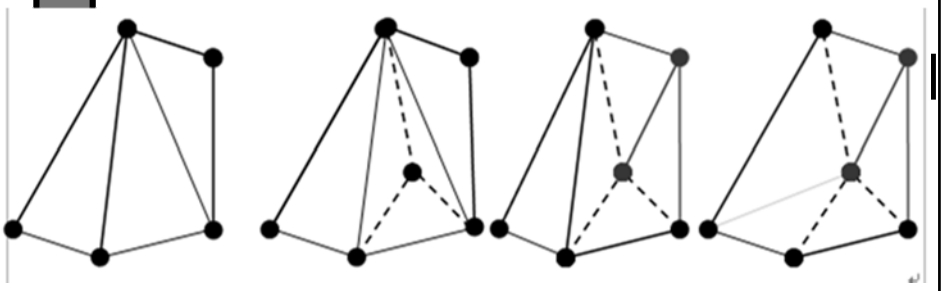


LegalizeEdge(p_r, , $\mathcal{T}$)

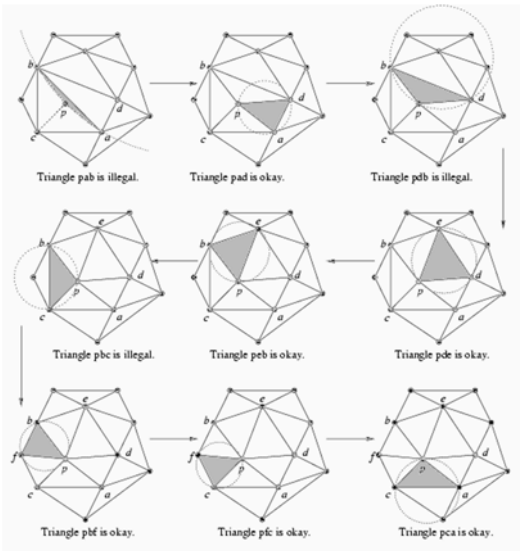
- 入力: p_r⇒挿入頂点, ⇒正当性検証の辺 $\mathcal{T}$ ⇒今までの三角形分割
- 出力: 新たな三角形分割$\mathcal{T}$

1. if ＝不正な辺
2. then
3. をフリップし、で置き換える
4. LegalizeEdge(p_r, , $\mathcal{T}$)
5. LegalizeEdge(p_r, , $\mathcal{T}$)

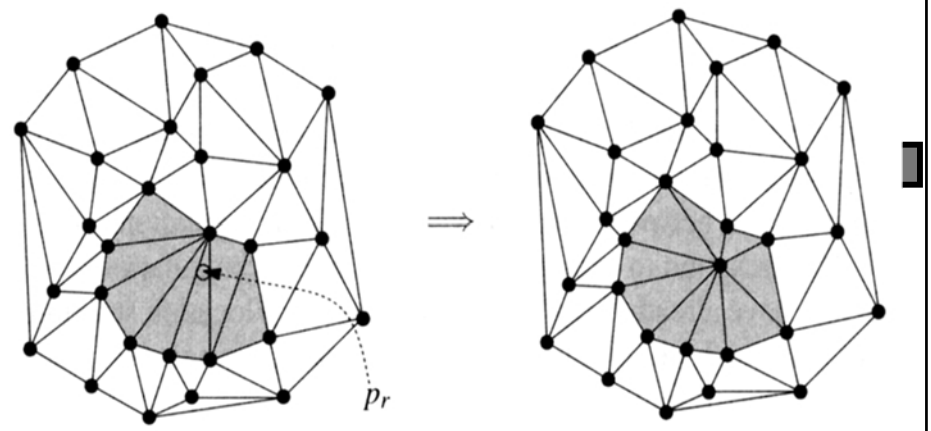
点の添加と三角形分割の変化-1



点の添加と三角形分割の変化-2

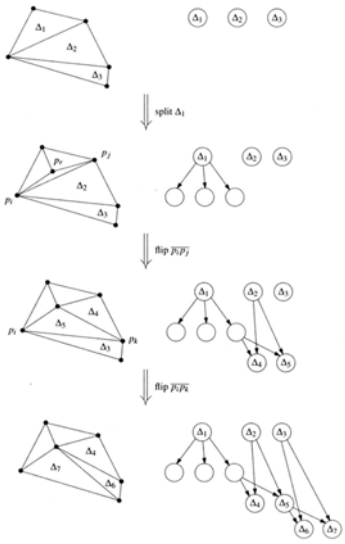


点の添加と三角形分割の変化-3



点の添加とデータ構造の変化

- 点p_rを含む三角形
  - 点位置決定問題
- 外点（葉）
  - 現在の三角形分割
- 内点
  - 以前に作られたが、すでに取り除かれた三角形



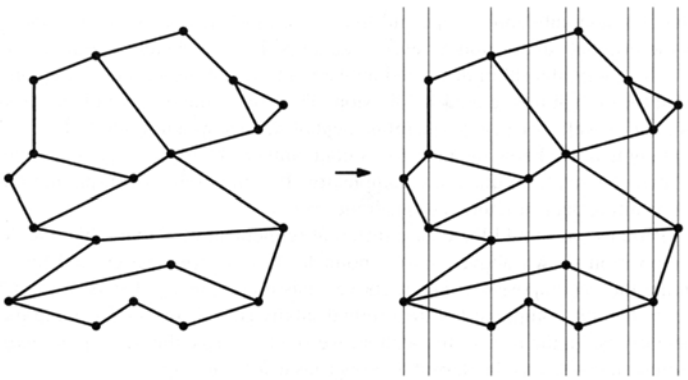
点位置決定問題

- 予め与えられた平面分割（平面の多角形領域への分割）に対し、質問点を含む多角形を迅速に見出す

1. 飛行中の航空機
   現在領空通過中の国家？
   ⇒現在地経緯度⇒地図
2. 走行中の自動車
   最も近いガソリンスタンド？
   ⇒車の現在地
   ⇒ガソリンスタンドのボロノイ図

多角形のスラブ分割

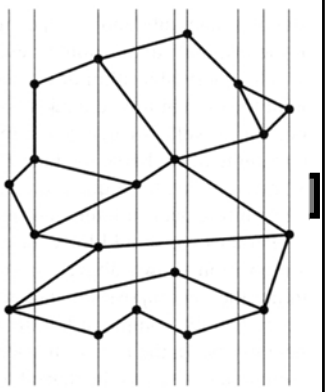
- スラブ（Slab）⇒多角形の各頂点を通る垂直な直線で平面を分割した時の帯



スラブ分割の考察

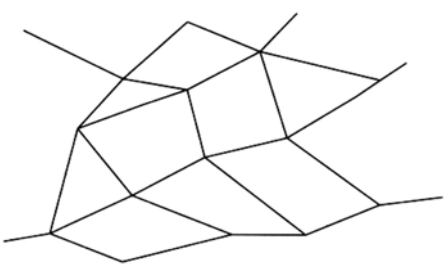
- Sをn個の辺を持つ平面領域分割とする

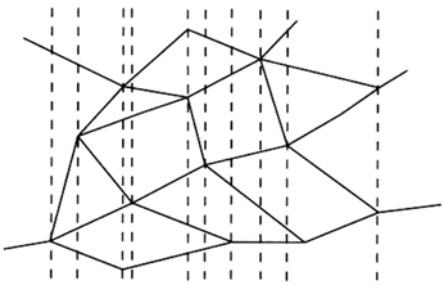
1. スラブ内にSの頂点は存在しない
2. スラブ内の辺は互いに交差せず、上下の順序がある
3. 2つの辺に囲まれたスラブ内の領域は、いずれもSの1つの面に属している
4. スラブ内の最も上と下の領域は有界でない（非有界）



Dobkin-Liptonアルゴリズム

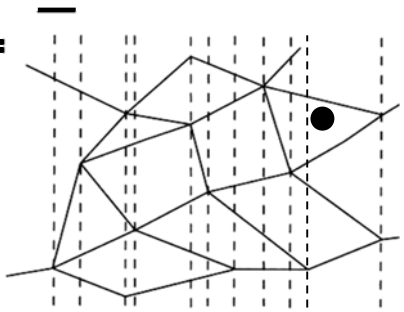
1. 多角形平面をスラブに分割する
2. 質問点qがどのスラブに入るかを探索する
3. スラブ内でどの線分の間にあるかを探索する



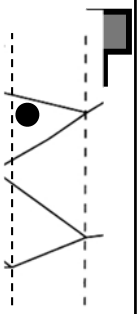


アルゴリズム

1. 平面領域分割の頂点のx座標値を蓄えた配列を生成



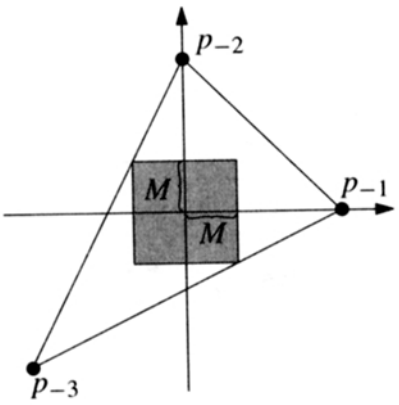
2. 質問点qのx座標値を用いて、上記の配列に関する2分探索
⇒qを含むスラブを見出す



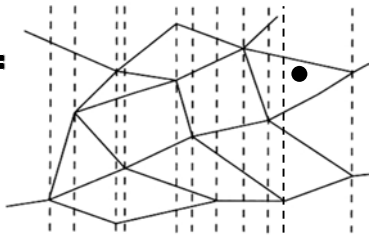
3. 見つけたスラブ内の辺の上下関係から、2分探索を行い、qのある面の直ぐ上下の線分を見出す

点p_{-1}, p_{-2}, p_{-3}の選び方

1. M←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. $p_{-3}=(-3M, -3M)$



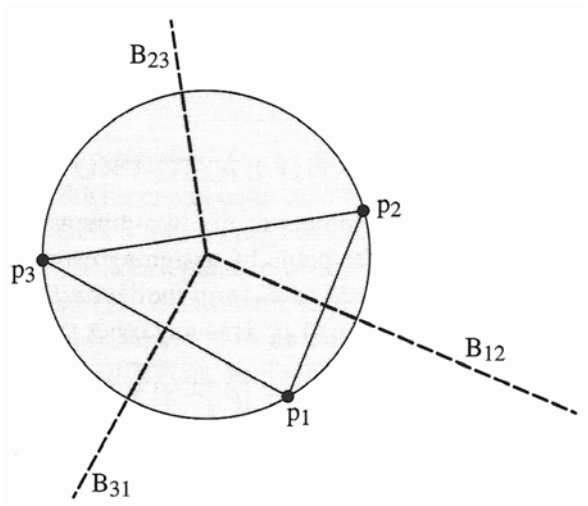
計算量



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

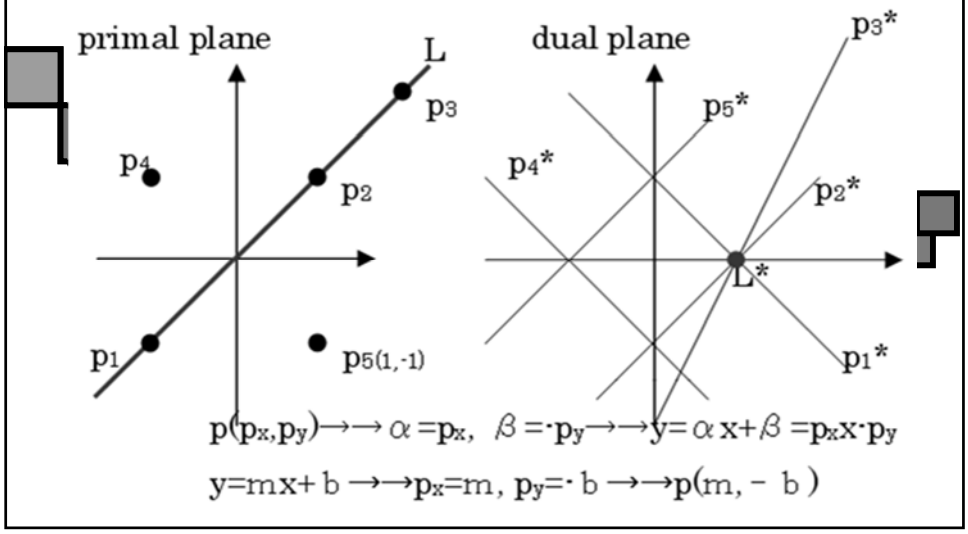
- 具体の例:
  - 平面上1点p(x, y)→2ﾊﾟﾗﾒｰﾀ（座標x, y）
  - 平面上1直線y=αx+β→2ﾊﾟﾗﾒｰﾀ（傾きα、切片β）
  - We can map a set of points to a set of lines, and vice versa, in a one-to-one manner.
  - 色々な映写方法がある
    - 直線⇔点
      - 1直線とその上の3点⇔同じ点を通る3本直線

双対変換(duality transform)

- 方法:

1. 点→直線
   - 点p:=(p_x, p_y) → 直線p*: y=p_xx - p_y
2. 直線→点
   - 直線L: y=mx + b → 点L*:=(m, -b)

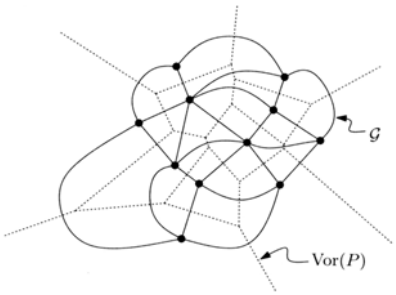
双対変換(duality transform)



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

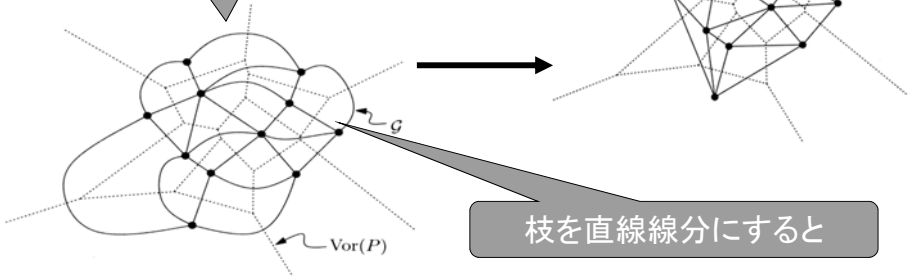


ドローネグラフ(Delaunay graph)

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

Pのドローネグラフ$Dg(P)$

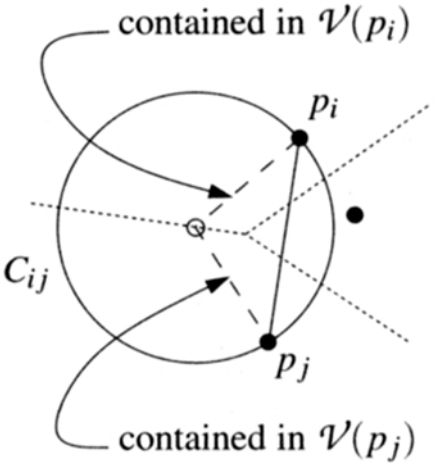
Pのドローネ三角形分割



枝を直線線分にすると

ドローネ辺 p_ip_j 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

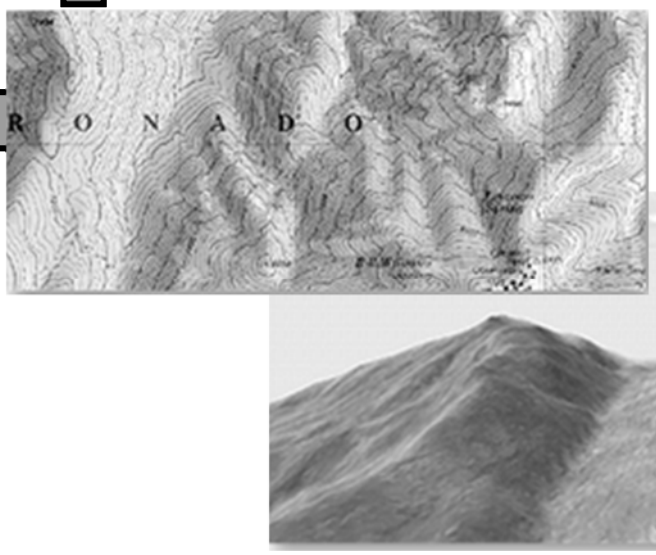
- Pを平面上の点集合とし、TをPの三角形分割とする

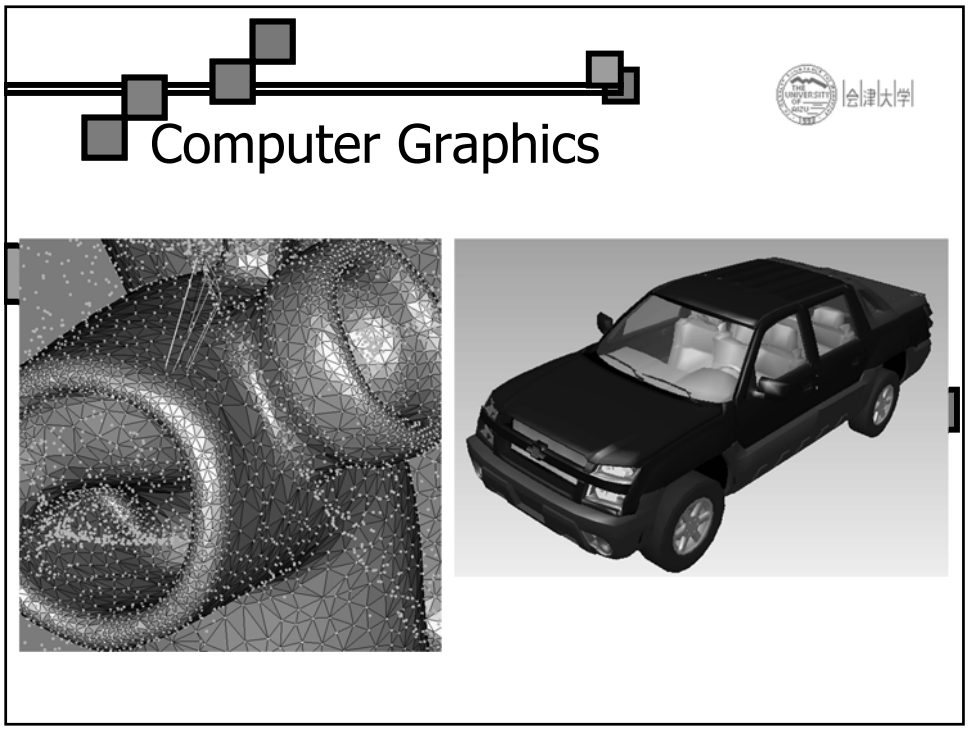
1. 3点p_i, p_j, $p_k \in P$がドローネグラフの同じ面の頂点であるための必要十分条件は、p_i, p_j, p_kを通る円の内部にPの点を含まないこと
2. 2点p_i, $p_j \in P$の線分 p_ip_j がPのドローネグラフの辺であるための必要十分条件は、Pの他の点を含まない、p_i, p_jを通る円が存在すること
3. TがPのドローネ三角形分割であるための必要十分条件は、Tの任意1つ三角形の外接円の内部にPの点を含まないこと
4. Pの三角形分割Tが正当であるための必要十分条件は、TがPのドローネ三角形分割であること
5. Pの角度最適な三角形分割⇔Pのド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

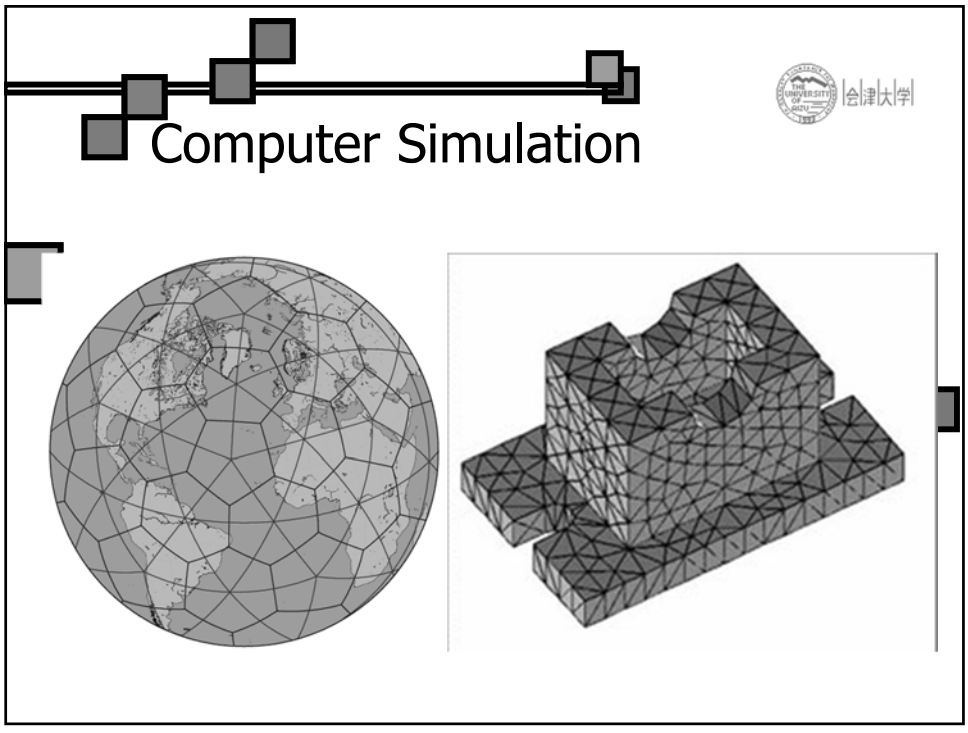
応用

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

Geographical Information System




Computer Graphics


Computer Simulation