計算幾何学
Computational Geometry

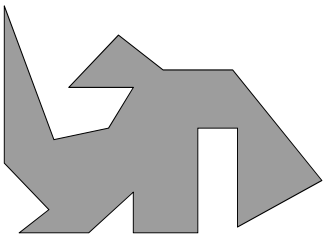
第七章　多角形の三角形分割
Polygon Triangulation

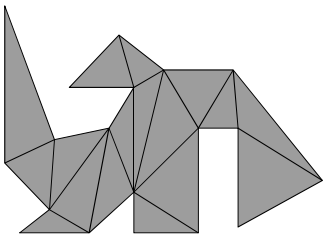
内容

- 定義
  - 多角形の三角形分割
- 問題の提起
  - 美術品監視→美術館問題
- 三角形分割のプロセス
  - 単純多角形 (simple polygon)
  - 単調多角形 (monotone polygon)
  - 三角形 (triangle)

多角形の三角形分割

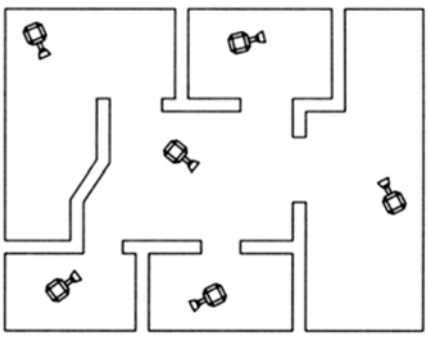
- Pをn頂点の単純な多角形とする
- Pの三角形分割
  - 交差しない対角線の極大集合によって、多角形を三角形に分割したもの
  - 対角線：Pの2頂点をPの内部だけを通って結ぶ開線分





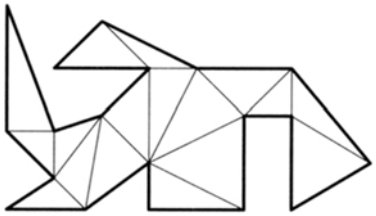
美術館問題(art gallery problem)

- 全館貴重な展示品を漏れなく監視するには
  →必要なカメラ台数？



問題の一般化

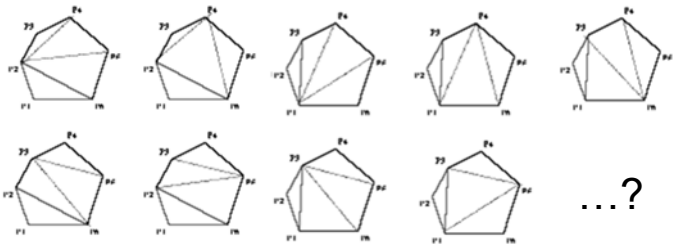
- 多角形*P*の内に、最小個数の点集合*G*を選び、*P*内のどの点も*G*の少なくとも一つの点から見えるようにする
- *P*内の2点が互いに見える(visible)→この2点を結ぶ線分の全体が*P*内にある
- *P*を三角形に分割
  - 一つ三角形領域に一台カメラ→全域漏れなく見える



三角形分割の考察

- 多角形を対角線で区切り、三角形に分割。
- 対角線同士は互いに交わらないように。

1. 三角形分割は常に存在する？ **→Y**
2. 多角形に何個の三角形がある？ **→n-2**
3. 一意に決まらない→何通りある？ **→$_{2n}C_n/(n+1)$**

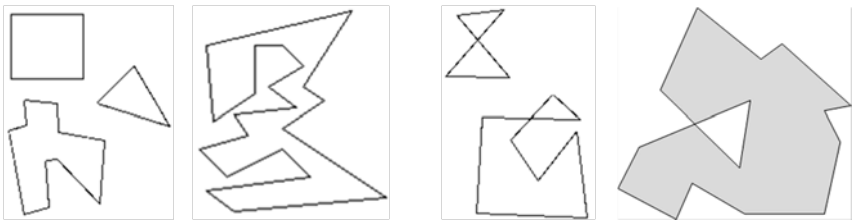


…?

単純多角形(simple polygon)

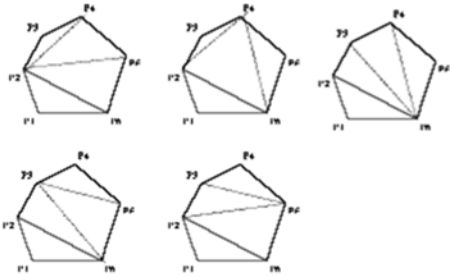
- 互いに<u>交差しない</u>閉じた多辺連続チェインに囲まれた領域
  - 平面を2つの互いに素な領域に分割する
  - 有界な内部領域＋非有界な外部領域
  - 有界領域内に穴がない

Simple polygons　　Non-simple polygons



対角線と三角形分割

- 対角線(diagonal)
  単純多角形の内部を通って、2頂点を結ぶ開線分
- 三角形分割(triangulation)
  交差しない対角線の極大集合（？）によって多角形を三角形に分割すること

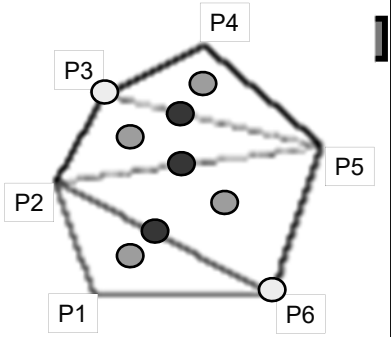


定理1

- どんな単純多角形も三角形分割が可能
- ***n*個の頂点**を有する単純多角形の任意の三角形分割にはちょうど***n*-2個の三角形**がある

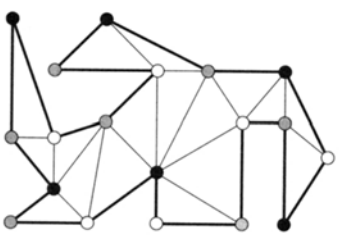
考察:カメラ設置位置と台数

- 三角形内→n-2
- 対角線上→$n/2$
- 頂点上→$n/3$



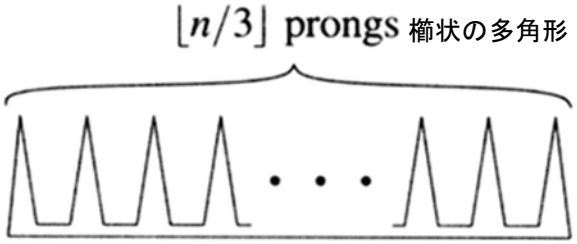
3-彩色(3-coloring)

- 単純多角形の各頂点に白、灰、黒のいずれかの色を割り当てる。但し、隣接の2頂点は必ず異なる色を付ける
- どの三角形も白・灰・黒の頂点を一つずつ持つ
- 何れの色のみにカメラを配置すれば、多角形全体の監視が可能
- 高々$\lfloor n/3 \rfloor$台のカメラが十分



最悪の例

- 頂点に配置されたカメラの監視範囲は接続している三角形に限らない→改善の余地？
- $n/3$台以下のカメラで任意の多角形を監視する可能性は？
- 答え⇒*No*

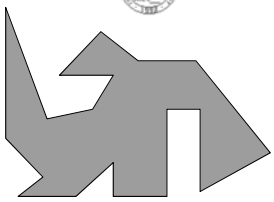


定理2（美術館定理）

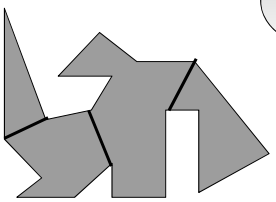
- n個の頂点を有する単純多角形に対して、多角形内の任意の点が少なくとも一台のカメラから見えるようにするのに、$\lfloor n/3 \rfloor$台のカメラが必要になることがあり、またその台数で常に十分である
  - $n/3$台は必要なカメラ台数の上限
  - 言い換えれば、$n/3$台以上を必要とする可能性はない。$n/3$台以下で監視できる時もある
  - 充分条件であるが、必要条件ではない

三角形分割のプロセス

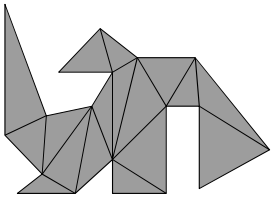
単純多角形(simple polygon)



単調多角形(monotone polygon)

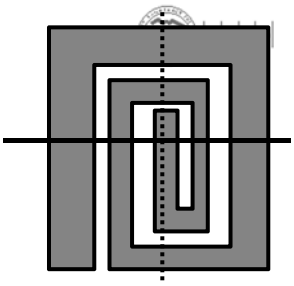


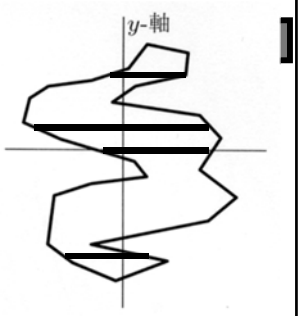
三角形(triangle)



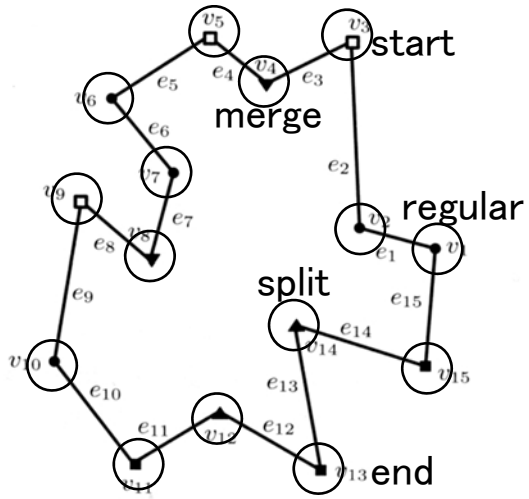
y単調多角形の特徴

- y軸と垂直な直線（水平線）に対して、多角形と交差する部分の線分が**連結**である
- 最も上の頂点から最も下の頂点まで、左側又は右側の境界の多辺連続チェインをたどる時、その走行方向は常に**下向き**又は**水平方向**だけ⇒**変曲点なし**
- 変曲点＝向き変化発生の点



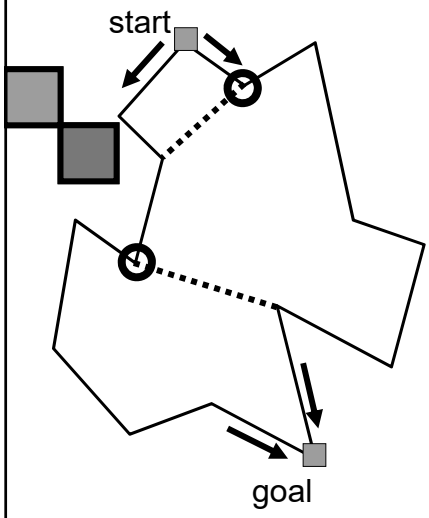


5種類の頂点



頂点種類	左右隣接2頂点	内角
□ 出発点	全て下方	$<\pi$
■ 最終点	全て上方	$<\pi$
● 普通点	上・下方各一点	$<\pi$ $>\pi$
▲ 分離点	全て下方	$>\pi$
▼ 統合点	全て上方	$>\pi$

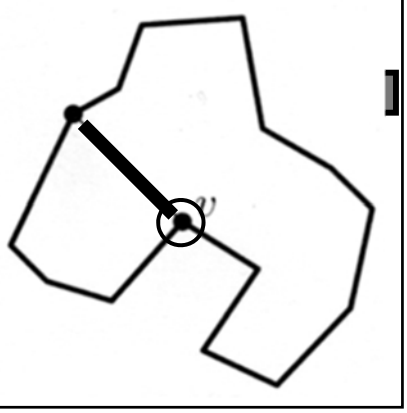
y単調な多角形を求める



- 多角形の最も上の頂点から最も下の頂点へたどっていくとき、上下向きが変化する頂点（**変曲点**）を取り除けば良い
- 取り除く⇒変曲点から**対角線**を引く

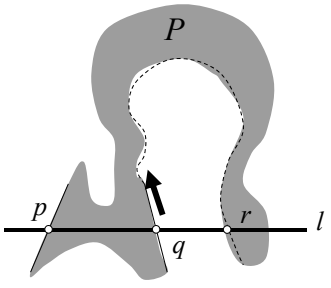
単純多角形→単調多角形

- 変曲点(turn vertex)
  - 多角形の境界をたどる時、向きが変化する頂点
    - 出発点、最終点
    - 分離点、統合点
- y単調に分割の目的
  - 変曲点の除去
    （分離点と統合点）
- y単調に分割の方法
  - 対角線を添加



y単調多角形の性質

- 分離点も統合点も持たない多角形。
- 論題:y単調でないPは分離点か統合点を持つ。

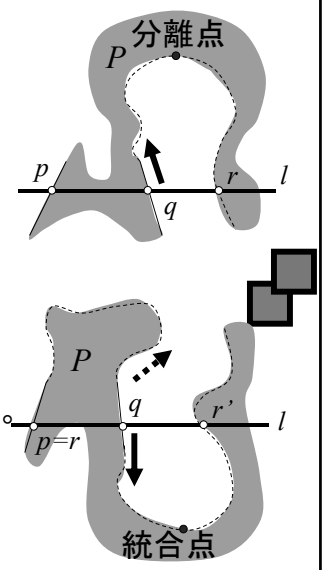


- 証明:

1. Pはy単調でないので、Pと交差する部分が2つ以上の線分に分かれるような水平線lが存在する。
2. このとき最も左の線分の左端点をp, 右端点をqとする。
3. qから出発して、Pの内部を左に見ながらPの境界線を辿ると、ある点rで境界線はlと再び交差するはずである。

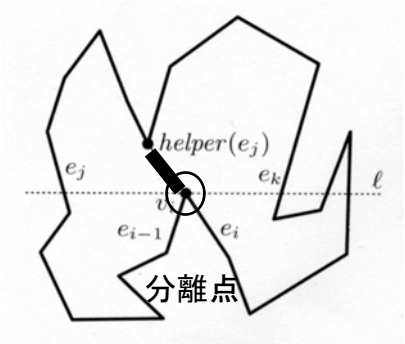
証明（続き）

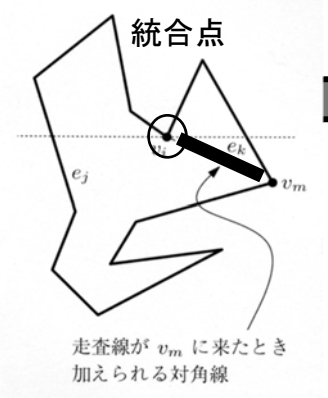
4. $p \neq r$のとき
   - qからrまでの道のりで最も高いところにある頂点は分離点でなければならない。
5. $p = r$のとき
   - qから出発して、Pの内部を右に見ながらPの境界線を辿る。このとき境界線がlと交差する点をr'とする。
   - もし$r'=p$なら、Pの境界線はlと2回しか交差しないことになるため、$r' \neq p$である。
   - このときqからr'までの道のりで最も低いところにある頂点は、統合点でなければならない。



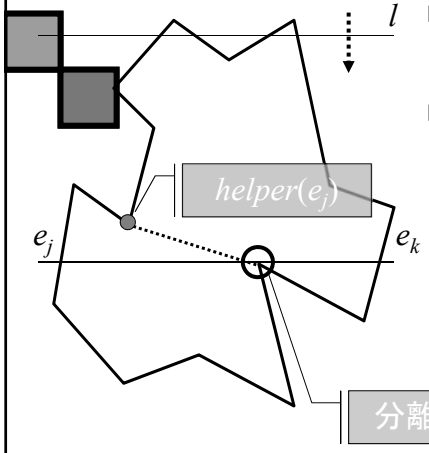
分離点と統合点の除去

- 分離点から上の頂点に向かう対角線を引く
- 統合点から下の頂点に向かう対角線を引く
- 対角線の具体的な引き方？





分離点の処理

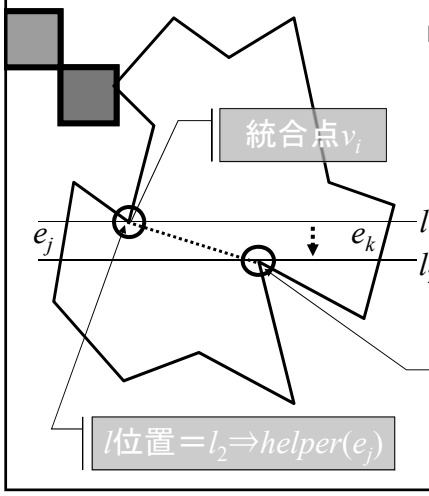


- 仮想的な走査線lを上から下へと動かしていく。
- lが分離点v_iに到達したとき、lからの距離が最小でv_iより上にある**頂点**と、v_iとを結ぶ。

⇩

辺e_jのヘルパー

統合点の処理



- lが統合点v_iに到達したとき、lからの距離が最小でv_iより下にある**頂点**と、v_iとを結ぶ。

⇩

v_iの次に
e_jのヘルパーになる頂点

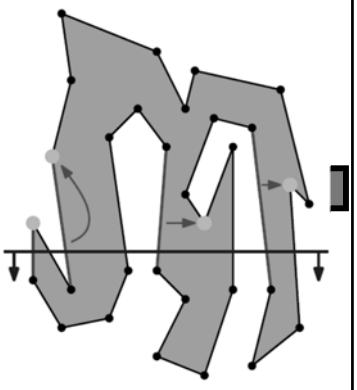
平面走査法

基本的な考え方

- $v_1, v_2, \ldots, v_n$　逆時計回り順に並べた頂点
- $e_1, e_2, \ldots, e_n$　辺集合、$e_i = \overrightarrow{v_i v_{i+1}}$、$e_n = \overrightarrow{v_n v_1}$
- イベント点：　多角形の頂点
- 目的：
  1. 各分離点からその上にある頂点へ対角線を引く
  2. 各統合点からその下にある頂点へ対角線を引く

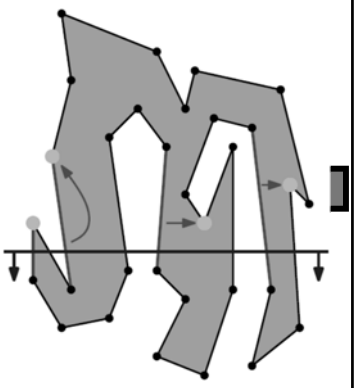
走査線状態

- 走査線と交差する辺の集合
- これらの辺の右側に多角形がある
- 付随の*Helper*の頂点と共に左から右へ蓄えられる



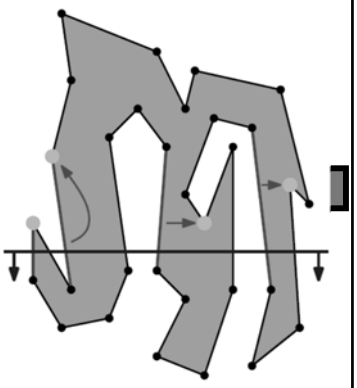
辺のヘルパー

- 辺eのヘルパー頂点vの特徴
  - 走査線より上に位置する一番低い頂点v
  - eとvの間を連結する水平線分は多角形内にある
  - 唯一・固定ではなく、走査に伴い変化する



データ構造

- 走査線状態
  - すべての辺（右に多角形がある）とその*Helper*を左から右へ2分探索木の外点に蓄える
- イベント
  - 頂点
  - y座標で並び替える
  - リスト、キュー、配列に保存



アルゴリズム

MakeMonotone(P)

- 入力：2重連結辺リストDに蓄えられた単純多角形P
- 出力：Dに蓄えられるPを単調多角形へ分割した結果

1. y座標値をキーとして、プライオリティキューQを構成する（y座標値が同様な2点について、x座標値が小さい方が優先）
2. 走査線と交差するPの辺とそのヘルパーを蓄える2分探索木Tを初期化
3. while $Q \neq$ empty
4. do Q の一番上から順番に v_i を取り出す
5. v_iの種類に応じて処理を行う

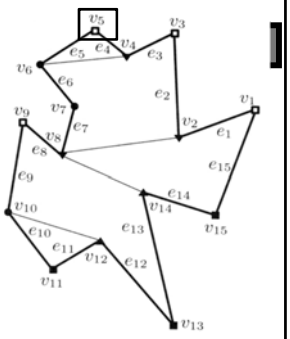
出発点の処理

HandleStartVertex(v_i)

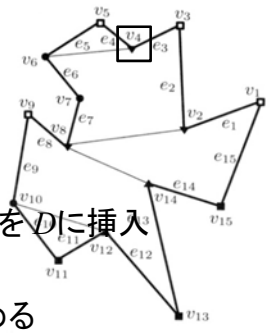
1. e_iをTに挿入
2. e_iのヘルパー$helper(e_i)=v_i$とする

v_5

- e_5をTに挿入
- e_5のヘルパー「$helper(e_5)$」＝v_5



統合点の処理



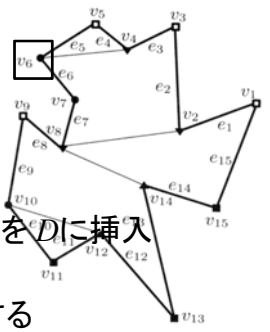
HandleMergeVertex(v_i)

1. if $helper(e_{i-1})$=統合点
2. 　　then v_iと$helper(e_{i-1})$ を結ぶ対角線をDに挿入
3. e_{i-1}をTから削除
4. Tを探索して、v_iのすぐ左の辺e_jを求める
5. if $helper(e_j)$=統合点
6. 　　then v_iと$helper(e_j)$ を結ぶ対角線を Dに挿入
7. $helper(e_j) \leftarrow v_i$

v_4

- e_3のヘルパー頂点v_3は統合点ではないため、対角線を引かなく、走査線lはe_3とTが交差しなくなるのでTからe_3を削除し、v_4のすぐ左の辺e_5を見つける。
- e_5のヘルパー頂点v_5は統合点ではないため、対角線を引かなく、e_5のヘルパー頂点v_5をv_4に変更。

普通点の処理



HandleRegularVertex(v_i)

1. if Pの内部がv_iの右にある
2. 　then if $helper(e_{i-1})$=統合点
3. 　　then v_i と $helper(e_{i-1})$を結ぶ対角線をDに挿入
4. 　　e_{i-1}をTから削除
5. 　　e_iをTに挿入し、$helper(e_i)=v_i$ にする
6. 　else Tを探索して、v_iのすぐ左の辺e_jを求める
7. 　　if $helper(e_j)$=統合点
8. 　　then v_i と $helper(e_j)$を結ぶ対角線を Dに挿入
9. 　　$helper(e_j) \leftarrow v_i$

v_6

- $helper(e_5)=v_4$が統合点なので、v_6からv_4へ対角線を引く
- 走査線lはe_5と交差しなくなりe_6と交差するようになるため、e_5をTから削除し、代わりにe_6を挿入
- $helper(e_6)=v_6$

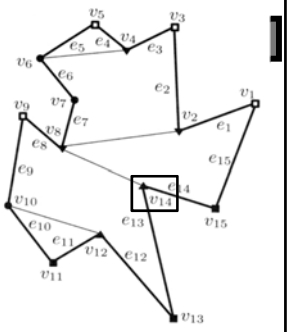
分離点の処理

HandleSplitVertex(v_i)

1. Tを探索して、v_iのすぐ左の辺e_jを求める
2. v_i と $helper(e_j)$を結ぶ対角線を Dに挿入
3. $helper(e_j) \leftarrow v_i$
4. e_iをTに挿入し、$helper(e_i)=v_i$ とする

v_{14}

- すぐ左の辺e_9の$helper(e_9)=v_8$とv_{14}を結ぶ対角線を加える
- e_9のヘルパー頂点をv_8からv_{14}に変更$helper(e_9)=v_{14}$
- e_{14}をTに挿入し、そのヘルパー頂点$helper(e_{14})=v_{14}$とする



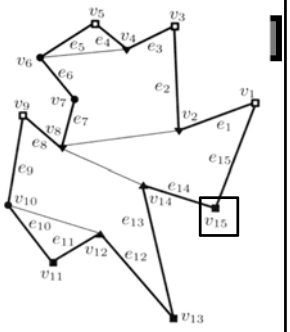
最終点の処理

HandleEndVertex(v_i)

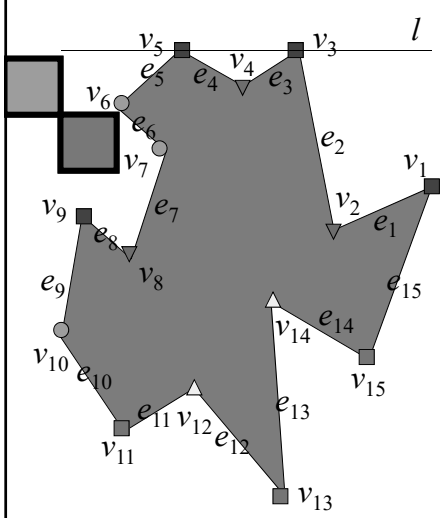
1. if $helper(e_{i-1})$=統合点
2. then v_iと$helper(e_{i-1})$ を結ぶ対角線を Dに挿入
3. e_{i-1}をTから削除

v_{15}

- $helper(e_{14})= v_{14} \neq$統合点なので、対角線を挿入する必要はない
- e_{14}をTから削除

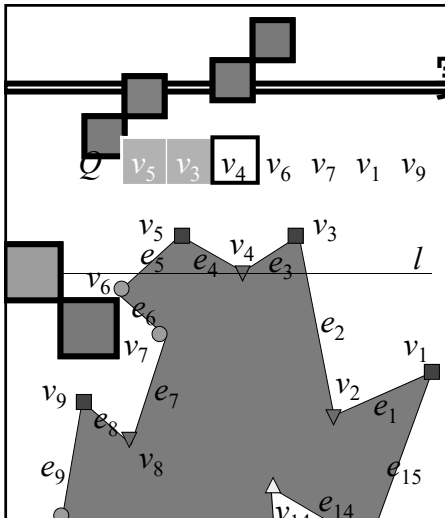


実行例

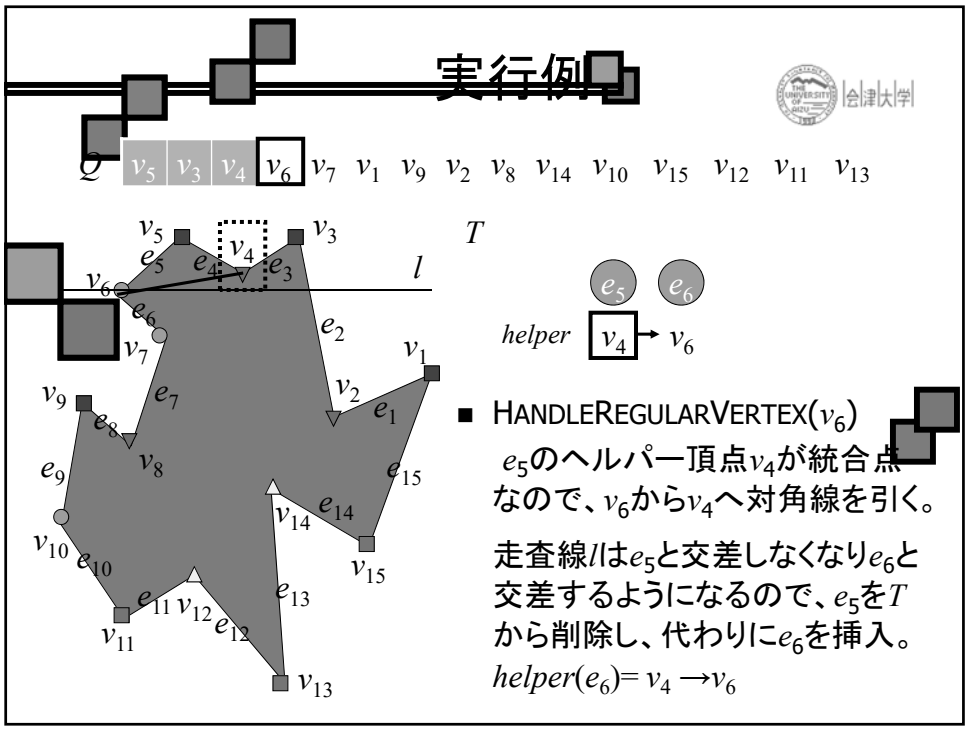


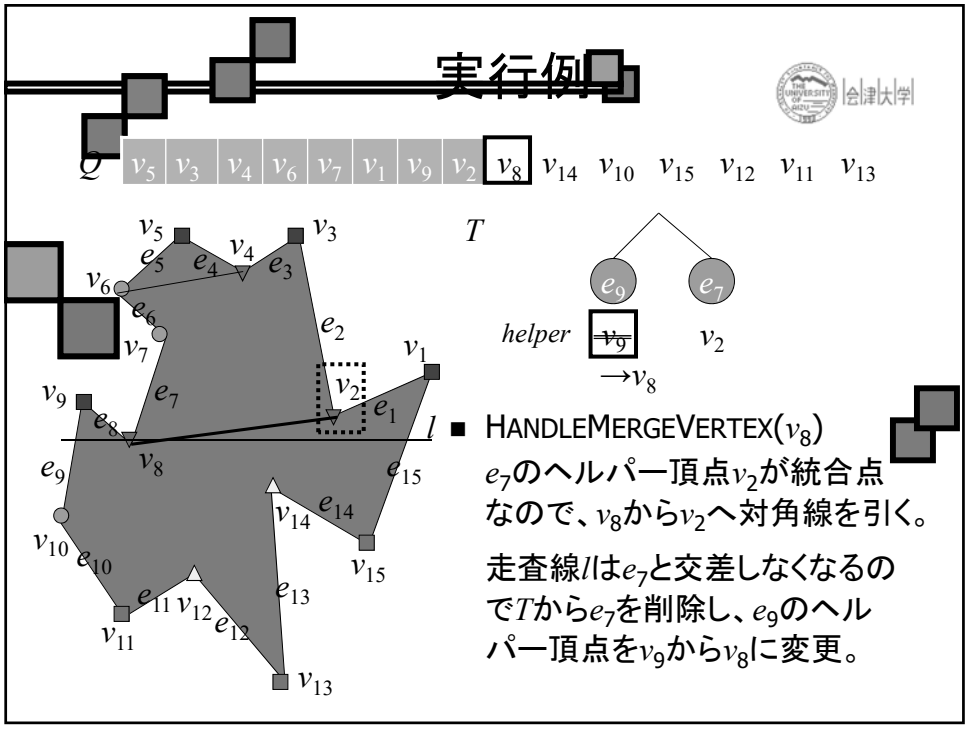
- HANDLESTARTVERTEX(v_5)
  e_5をTに挿入し、e_5のヘルパーをv_5とする。

実行例

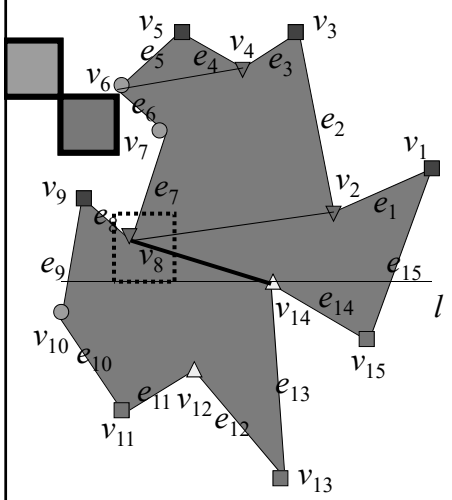


- HANDLEMERGEVERTEX(v_4)
  e_5のヘルパー頂点もe_3のヘルパー頂点も統合点ではないため、対角線を引かない。
  走査線lはe_3と交差しなくなるためTからを削除し、e_5のヘルパー頂点をv_5からv_4に変更。


実行例
Q
v5 v3 v4 v6 v7 v1 v9 v2 v8 v14 v10 v15 v12 v11 v13
T
e5 e6
helper v4 → v6
l
■ HANDLEREGULARVERTEX(v6)
e5のヘルパー頂点v4が統合点なので、v6からv4へ対角線を引く。
走査線lはe5と交差しなくなりe6と交差するようになるので、e5をTから削除し、代わりにe6を挿入。
helper(e6)= v4 →v6


実行例
Q
v5 v3 v4 v6 v7 v1 v9 v2 v8 v14 v10 v15 v12 v11 v13
T
e9 e7
helper v9 v2
→v8
l
■ HANDLEMERGEVERTEX(v8)
e7のヘルパー頂点v2が統合点なので、v8からv2へ対角線を引く。
走査線lはe7と交差しなくなるのでTからe7を削除し、e9のヘルパー頂点をv9からv8に変更。

実行例



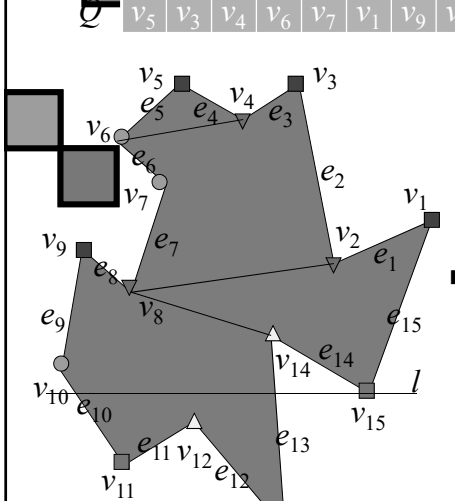
- HANDLESPLITVERTEX(v_{14})

  v_{14}とv_8（e_9のヘルパー頂点）を結ぶ対角線を引く。

  e_9のヘルパー頂点をv_8からv_{14}に変更。

  e_{14}をTに挿入し、そのヘルパー頂点をv_{14}とする。

実行例

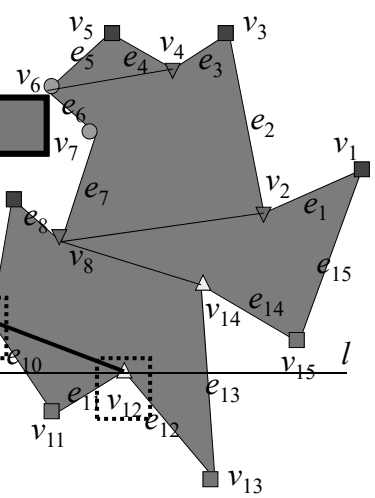


- HANDLEENDVERTEX(v_{15})

  e_{14}のヘルパー頂点v_{14}は統合点ではないため、対角線を引かない。

  e_{14}をTから削除。

実行例



- HANDLESPLITVERTEX(v_{12})
  v_{12}とv_{10}（e_{10}のヘルパー）を結ぶ対角線を引く。
  e_{10}のヘルパーをv_{10}からv_{12}に変更。
  e_{12}をTに挿入し、そのヘルパーをv_{12}とする。

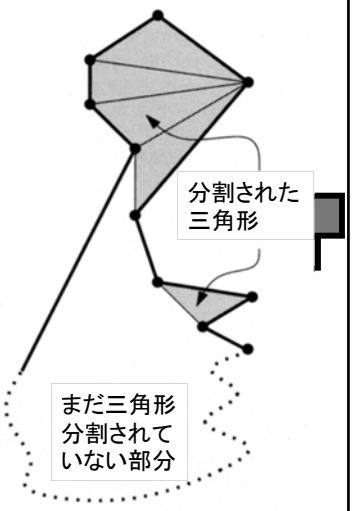
計算量

- プライオリティキューQの構成→$O(n)$時間
- 2分探索木Tの初期化→$O(1)$時間
- 1つ頂点に到達したときに行われる処理
  - Qに関する操作(1回)→$O(\log n)$時間
  - Tに関する質問(高々1回)、挿入、削除(1回)
    → $O(\log n)$時間
  - Dへの対角線の挿入(高々2本)→$O(1)$時間
- 1つイベントに対して$O(\log n)$時間
- n個の頂点に対して処理を行うので、全体での実行時間は$O(n\log n)$

単調多角形→三角形

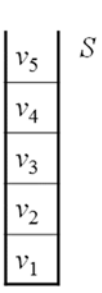
基本的な考え方

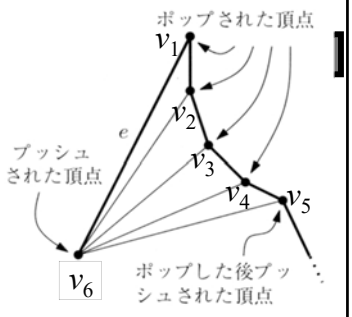
- y座標値の降順で一つずつ頂点を処理する
- 最高頂点から最低頂点まで、両側の境界をたどり、徐々に降りて行きながら、可能であれば、対角線を加え、三角形分割を行う



1つ頂点についての処理

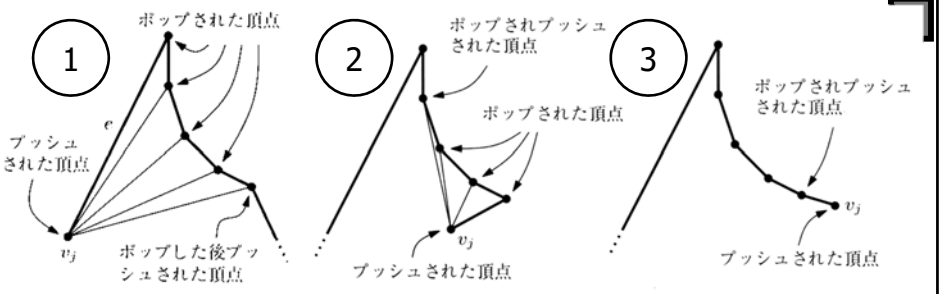
- 準備＝スタックS
  - ⇒ 既に出会ったが、また対角線を引ける頂点。
  - ⇒ まだ対角線が引かれていない頂点。
- 処理＝その頂点からS内にある頂点に向けて最大限に多数の対角線を引く
- v_6の処理⇒ S内v_1～v_5





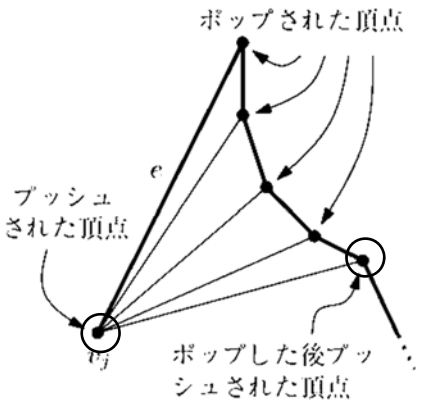
処理頂点v_jと相手頂点の位置関係

1. v_jがS中の相手頂点と異なる側
2. v_jがS中の相手頂点と同じ側（左）
3. v_jがS中の相手頂点と同じ側（右）



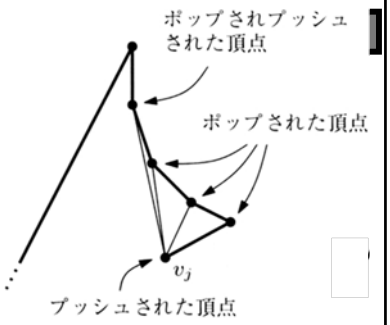
対角線の引き方ー①異なる側

- v_jからS中にあるすべての頂点（最高の頂点「eの上端点」除外）へ対角線を引く
- これらの頂点をSから全部取り出す
- v_jと最低の頂点をSに戻す



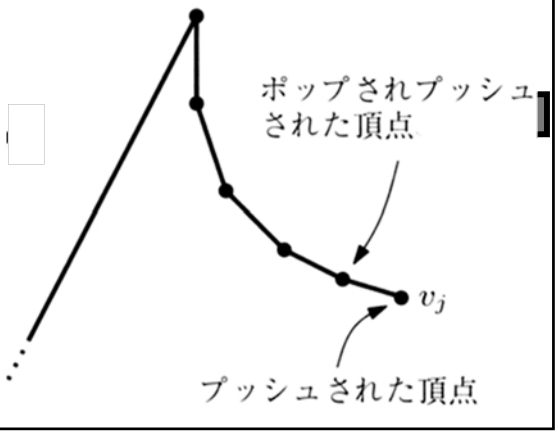
対角線の引き方ー②同じ側（左）

- v_jからS中にあるすべての頂点へ対角線を引けない可能性がある
- 最低頂点はすでにv_jと連結しているため、最低頂点の上にある複数の頂点を一つずつ取り出す
- 引ける頂点まで対角線を引く
- v_jと最後の対角頂点をSに戻す

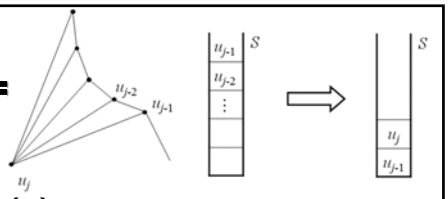


対角線の引き方ー③同じ側（右）

- v_jからS中にあるすべての頂点へ対角線を引けないため、取り出された頂点を再びSに戻す
- v_jをSに戻す



アルゴリズム



TriangulateMonotonePolygon(P)

- 入力：Dに蓄えられたy単調な多角形P
- 出力：Dに蓄えられたPの三角形分割

1. Pの左右側のチェイン上の頂点列を一つの系列に統合し、y座標値の降順に並び替える $u_1,\ldots,u_n$
2. u_1とu_2をスタックSに蓄える
3. for j=3 to n-1
4. 　if u_jとSの一番上の頂点が異なる側チェイン上にある
5. 　　then Sからすべての頂点を取り出す
6. 　　u_jと取り出されたそれぞれの頂点を結ぶ対角線をDに挿入（最後の頂点を除外）
7. 　　u_{j-1}とu_jをSに入れる

アルゴリズム続き

8. 　　else u_jとSの一番上の頂点が同じ側チェイン上にある
9. 　　Sから1つ頂点を取り出す
10. 　　u_jからの対角線がPの内部にある限り、Sから他の頂点を取り出す。これらの対角線を Dに挿入。取り出された最後の頂点をSに入れる
10. 　　u_jをSに入れる
11. 最初と最後の頂点を除いて、u_nからS上の全ての頂点への対角線を加える

計算量

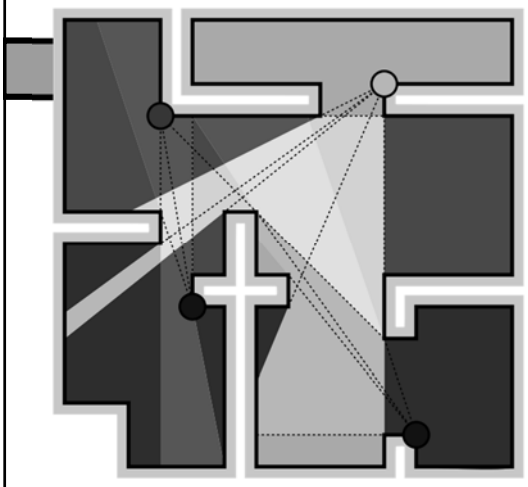
- u_1とu_2をスタックSにプッシュ→$O(1)$時間
- u_3~u_{n-1}の処理(n-3回の繰り返し処理)
  - スタックSへのプッシュの回数
    →1つ頂点の処理につき高々2回
  - スタックSからのポップの回数
    →プッシュの回数以下

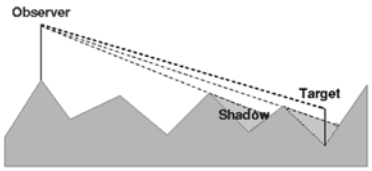
  ⇒$O(n)$時間
- n頂点を持つy単調な多角形は、線形時間で三角形分割できる。

応用

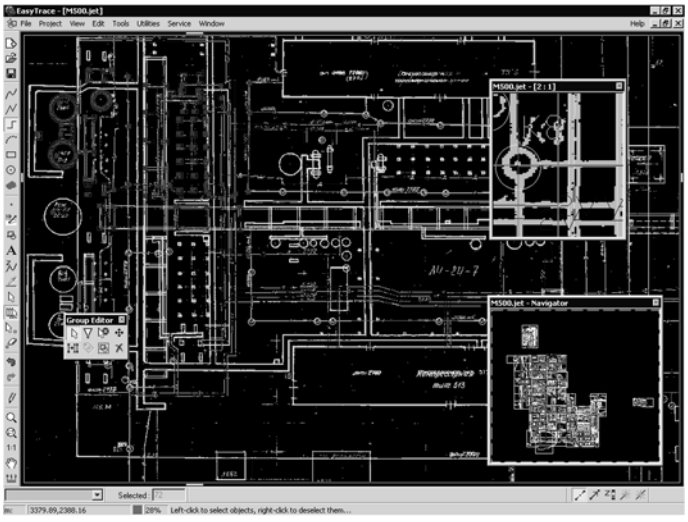
- 可視性と最短路問題
  - 美術館監視、警備員巡回、要塞防衛、刑務所警備
  - ロボット最適経路設計
- 有限要素法セル分割
- VLSI設計
- 画像処理
- コンピュータグラフィクス

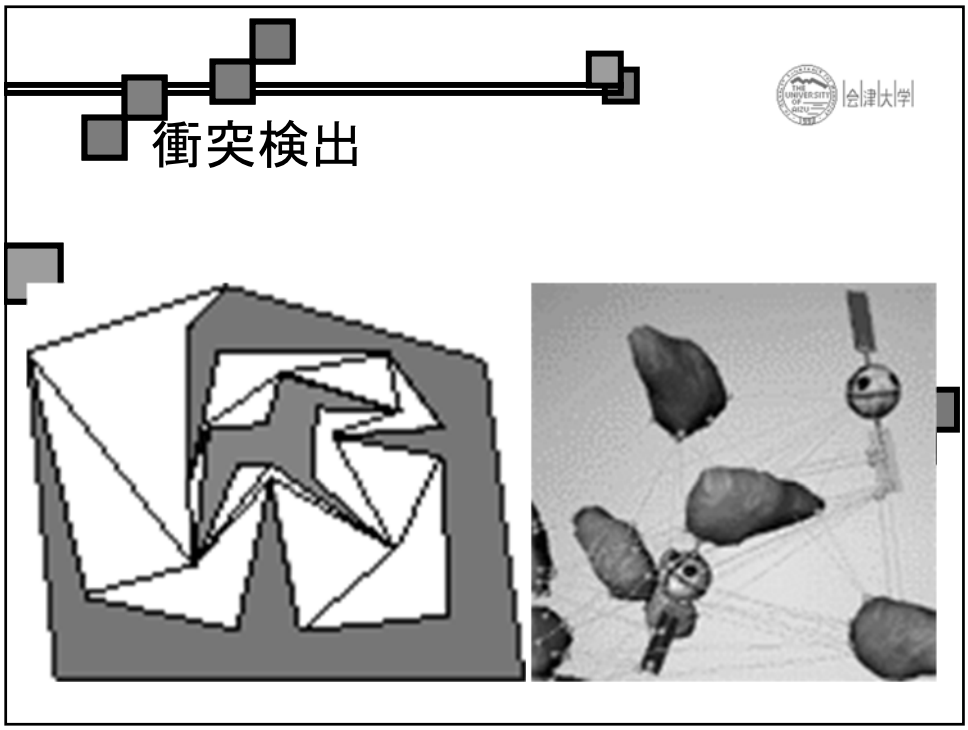
可視性問題

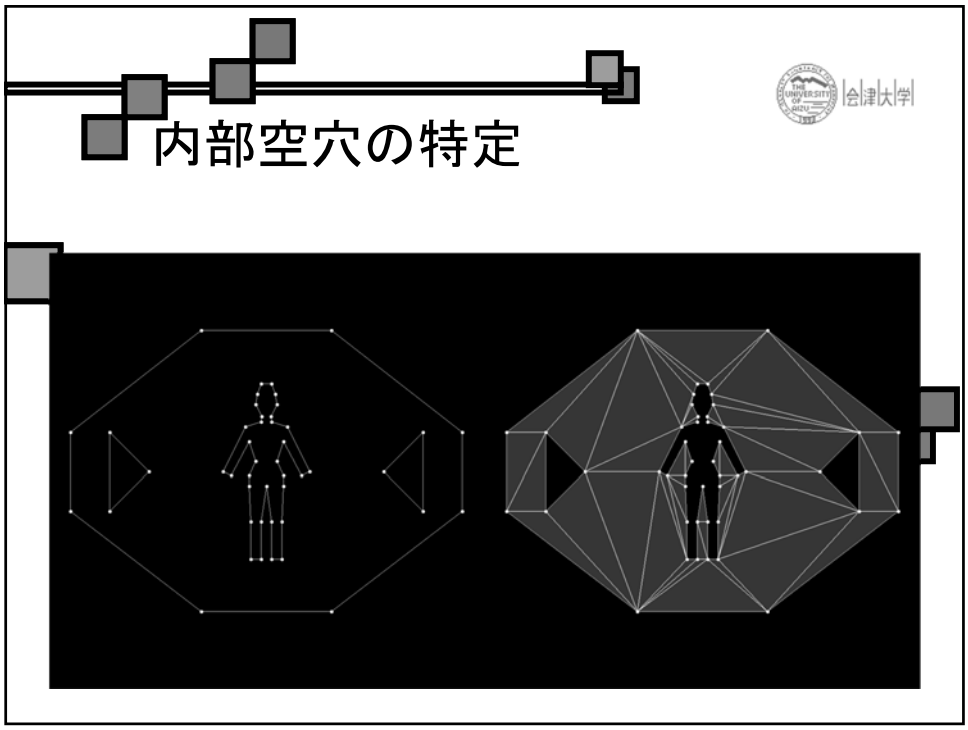


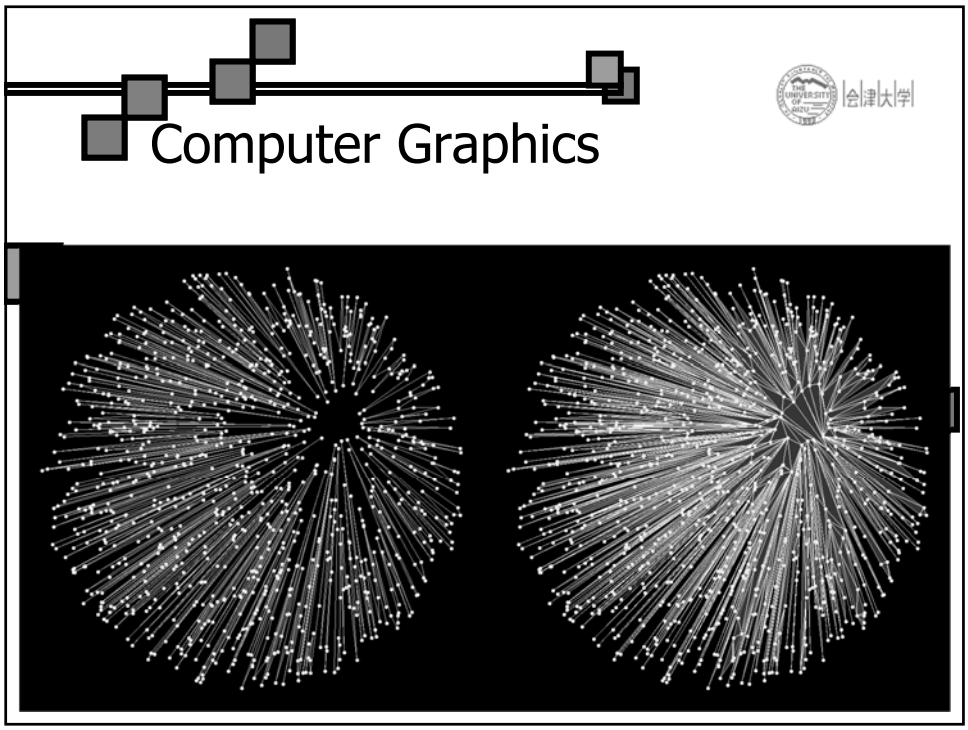

Observer
Target
Shadow

画像処理


EasyTrace - [M500.jet]
File Project View Edit Tools Utilities Service Window
Help
M500.jet - [2 : 1]
M500.jet - Navigator
Group Editor
Selected :
Left-click to select objects, right-click to deselect them...


衝突検出


内部空穴の特定


Computer Graphics
会津大学