情報メディア工学特論

等値面表示システム

2005/1/11

京都大学高等教育研究開発センター情報メディア教育部門

学術情報メディアセンター連携研究部門（兼任）

大学院工学研究科電気工学専攻（兼担）

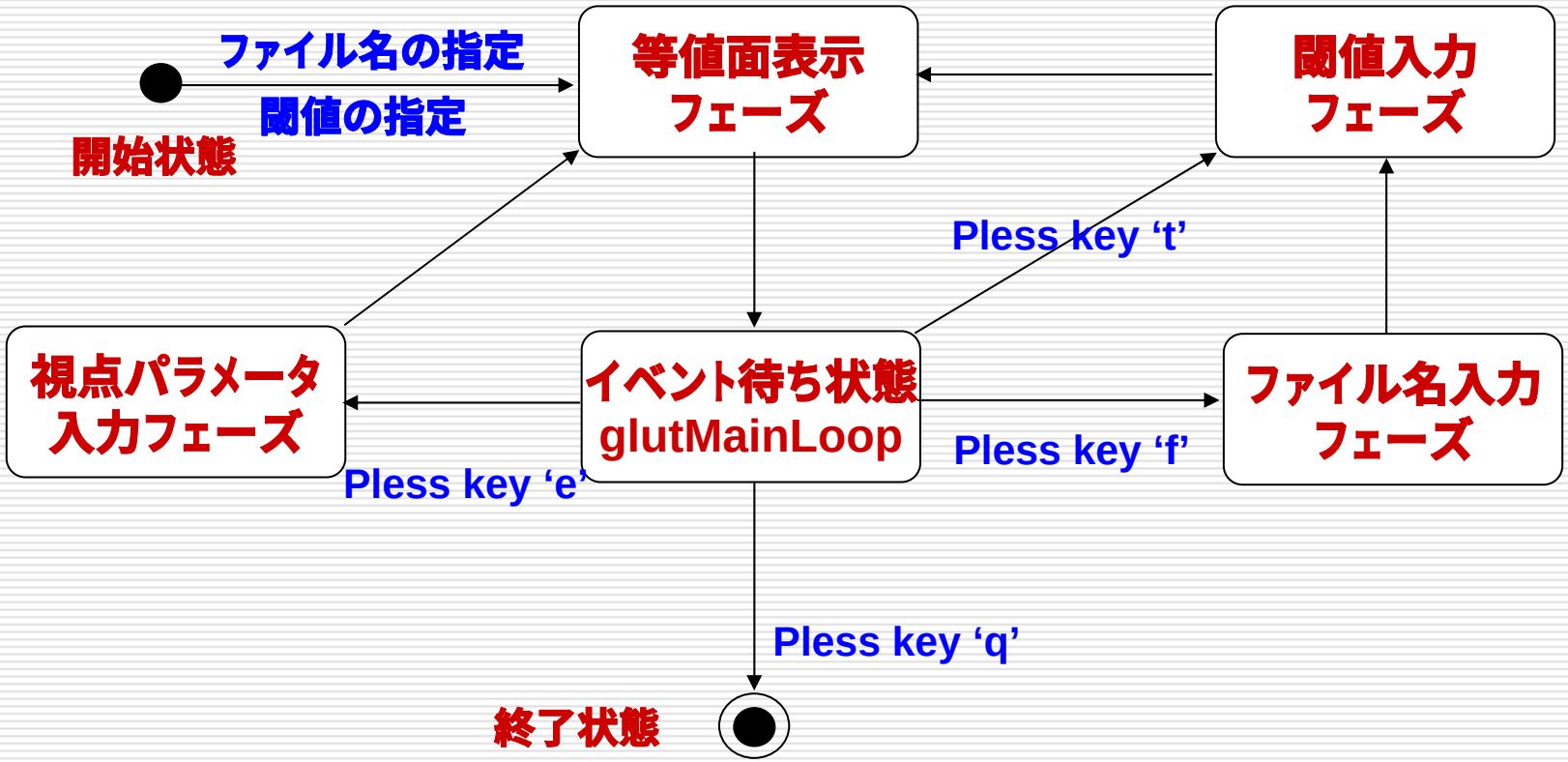
小山田耕二

コース概要 (2/2)

- 特異点ベース可視化技術
  - 渦可視化(11/16)
  - 等値面表示高速化、DT-MRI可視化(11/30)
- システム開発技術/OpenGL基礎
  - オブジェクト指向システム開発技術 (12/7)
  - 基本オブジェクト設計 (12/14)
- 可視化システム実装演習
  - 等値面表示システム(12/21)
  - ボリュームレンダリング表示システム (1/11)
  - 流線表示システム(1/18)

等値面生成システム概要

・状態遷移図



等値面表示アルゴリズム

(Lorensen, 1987)

各格子ごとに

各格子点で

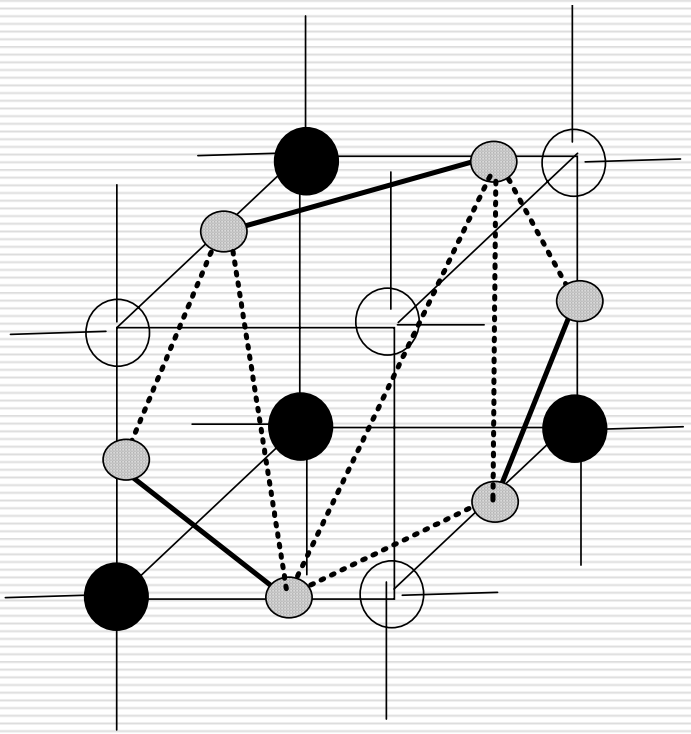
- スライス面より上方だと (+) と設定
- スライス面より下方だと (-) と設定

両端の格子点の符号が異なる稜線について

- 交差点を計算
- 交差点での数値データを定義

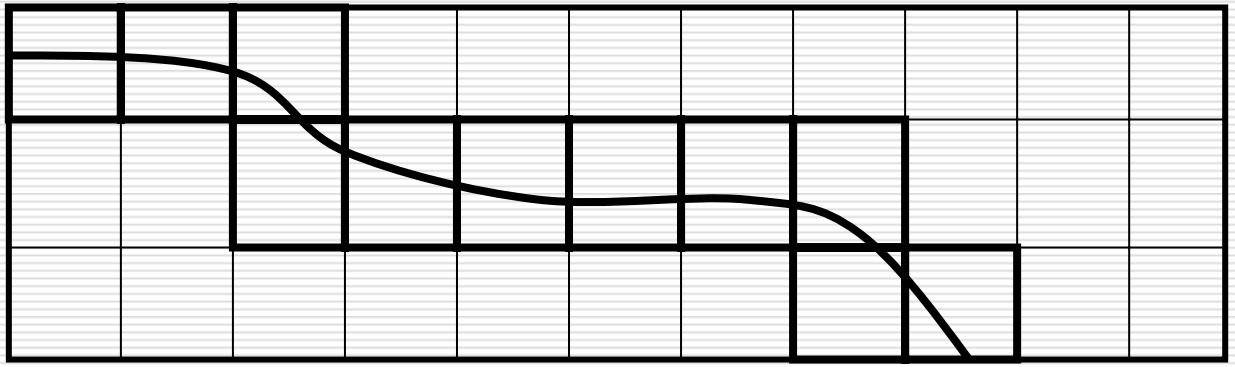
交差点を選んで3角形を定義

数値データ付き3角形を出力



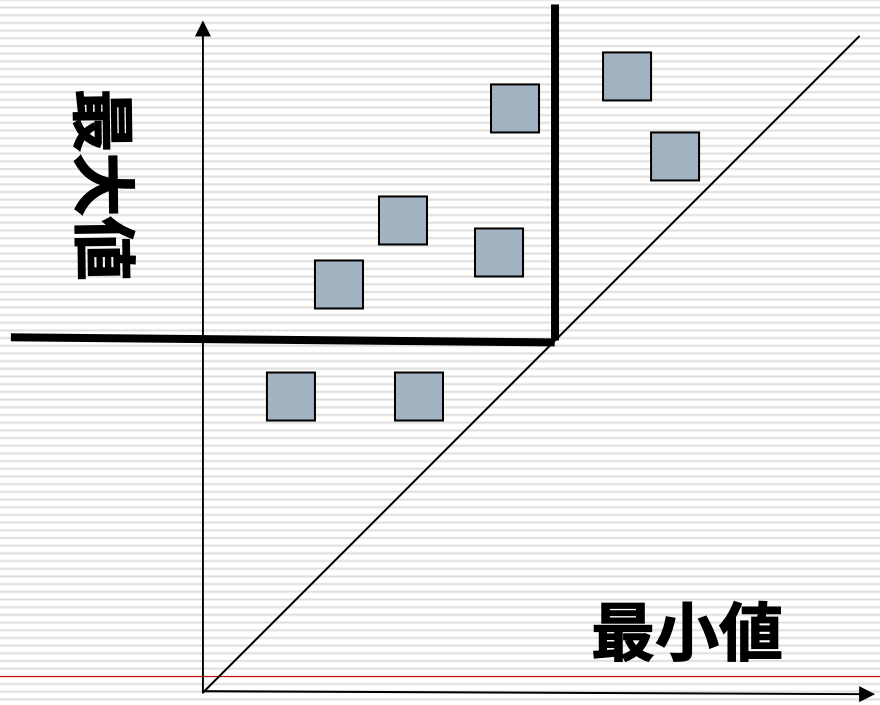
インデクスを使った高速探索

- 等値面と交差する格子を効率よく探索する
  - 空間ベース(8分木)
  - データ値ベース(区間木)
  - シードベース(極点グラフ)



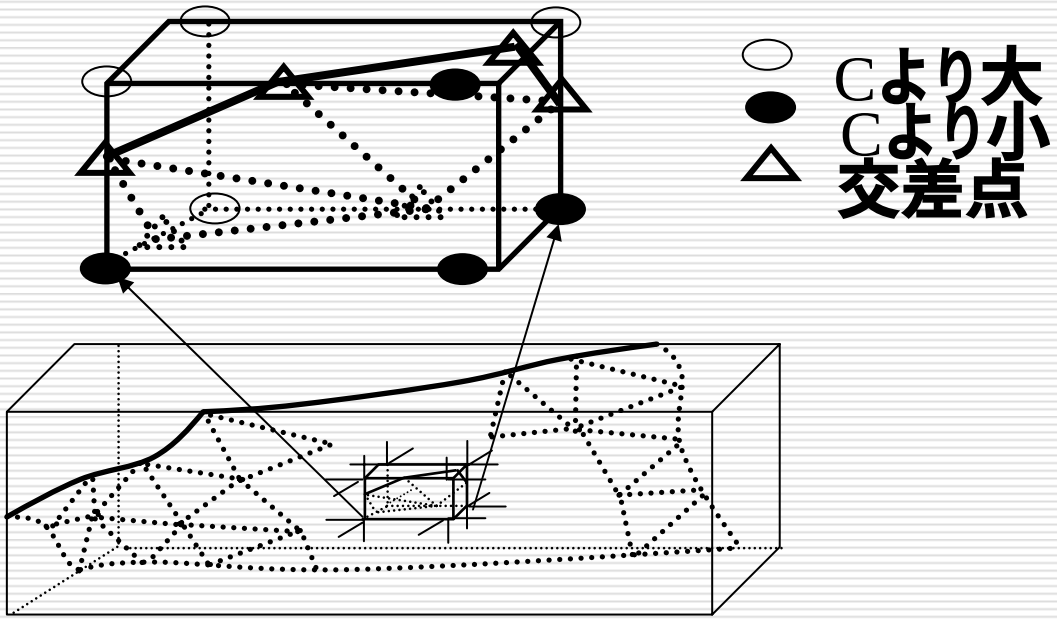
区間木の利用
(Shen, 1996)

- 格子点でのデータの(最小値,最大値)を使って格子を表現する



自己増殖的生成

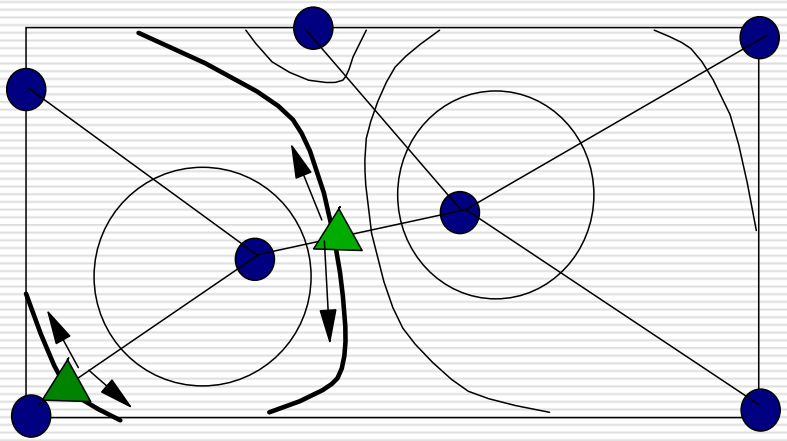
- 隣接情報を利用して等値面に交差する格子を特定する



極点グラフの利用
(Itoh and Koyamada, 1996)

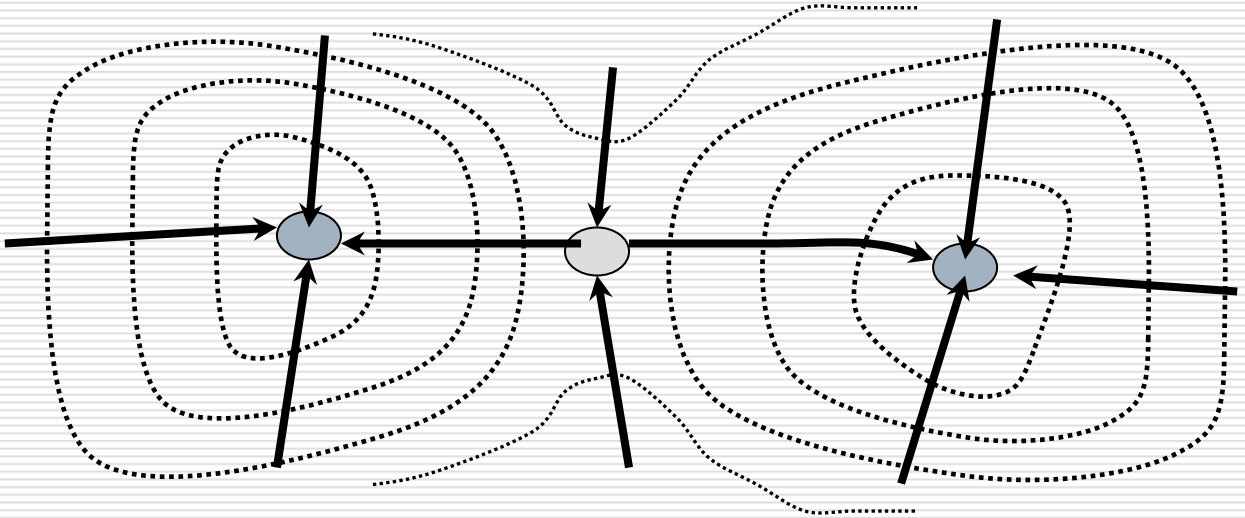
シード格子からの自己増殖型等値面生成
極点グラフの提案

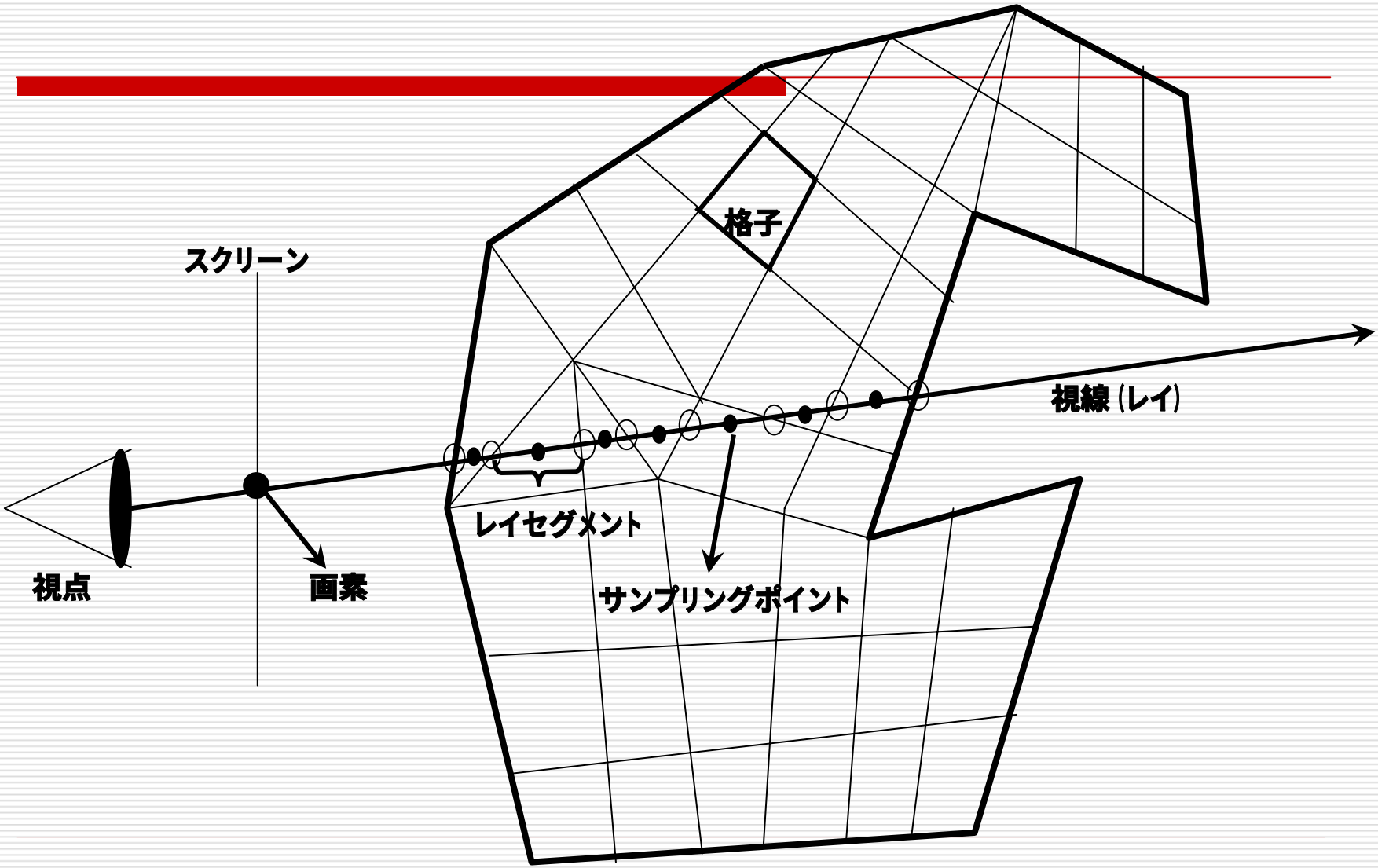
- **シード格子の特定を高速化**
- **多くの極値を持つ場合、極点グラフ生成がボトルネック**



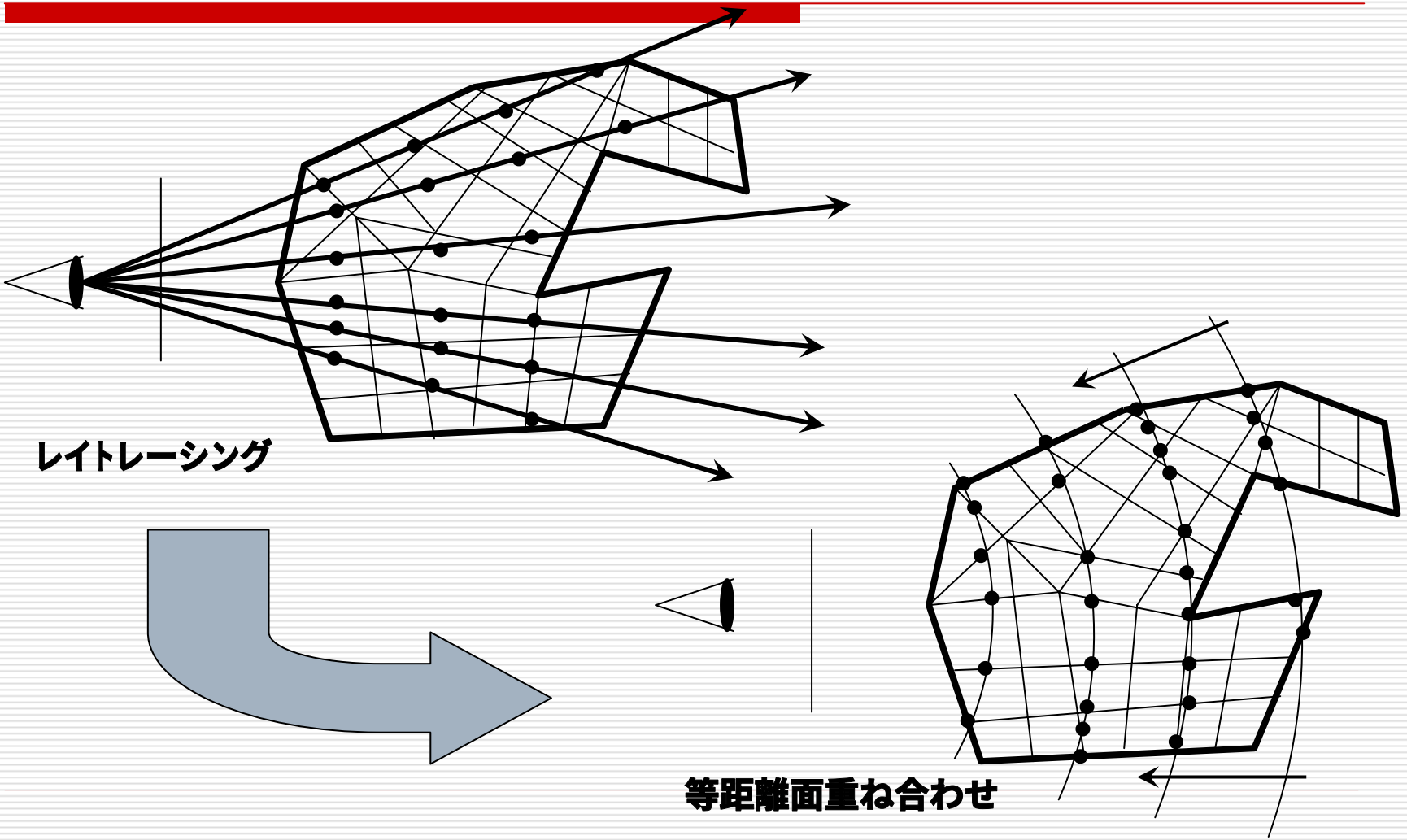
コンタツリー(C.L.Bajaj,1997)

- スカラ場の勾配ベクタに特異点理論を適用
  - 極大点、極小点、鞍点の探索
  - 上記特特異点をベクタ線で接続する



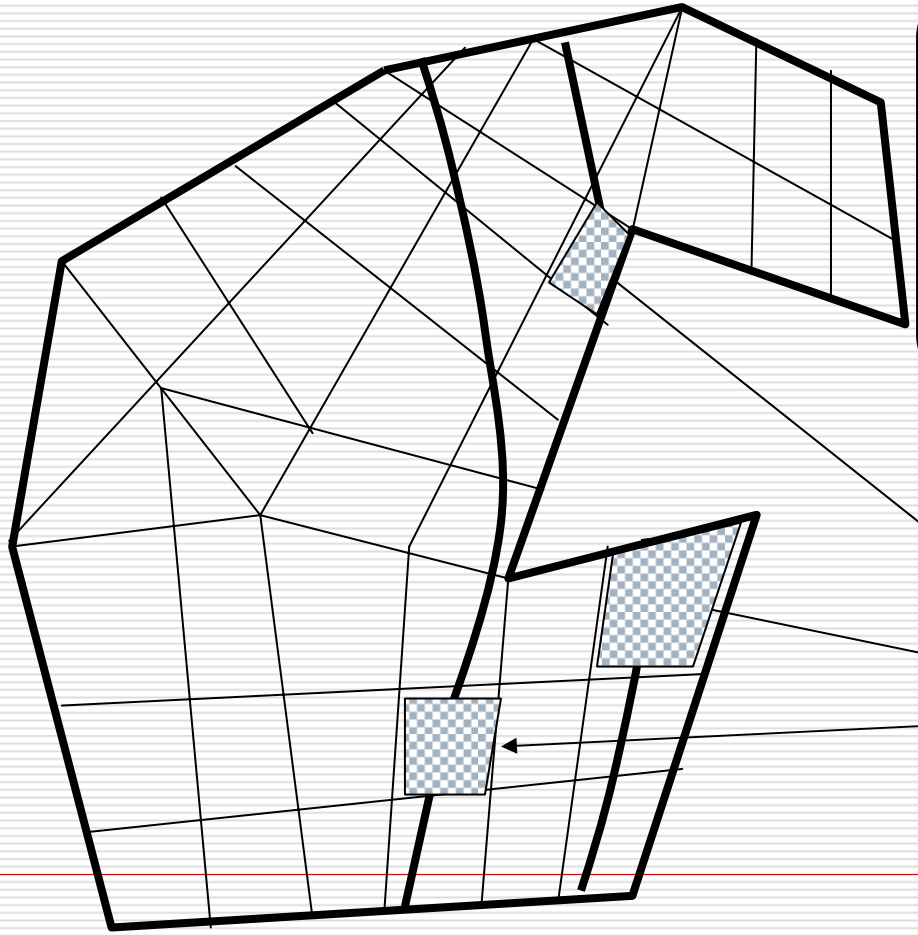

格子
スクリーン
視線（レイ）
レイセグメント
視点
画素
サンプリングポイント

等距離面重ね合わせによるボリュームレンダリング
(Koyamada and et. al, 1992)



シード格子の特定

(Speray and Kennon, 1990)

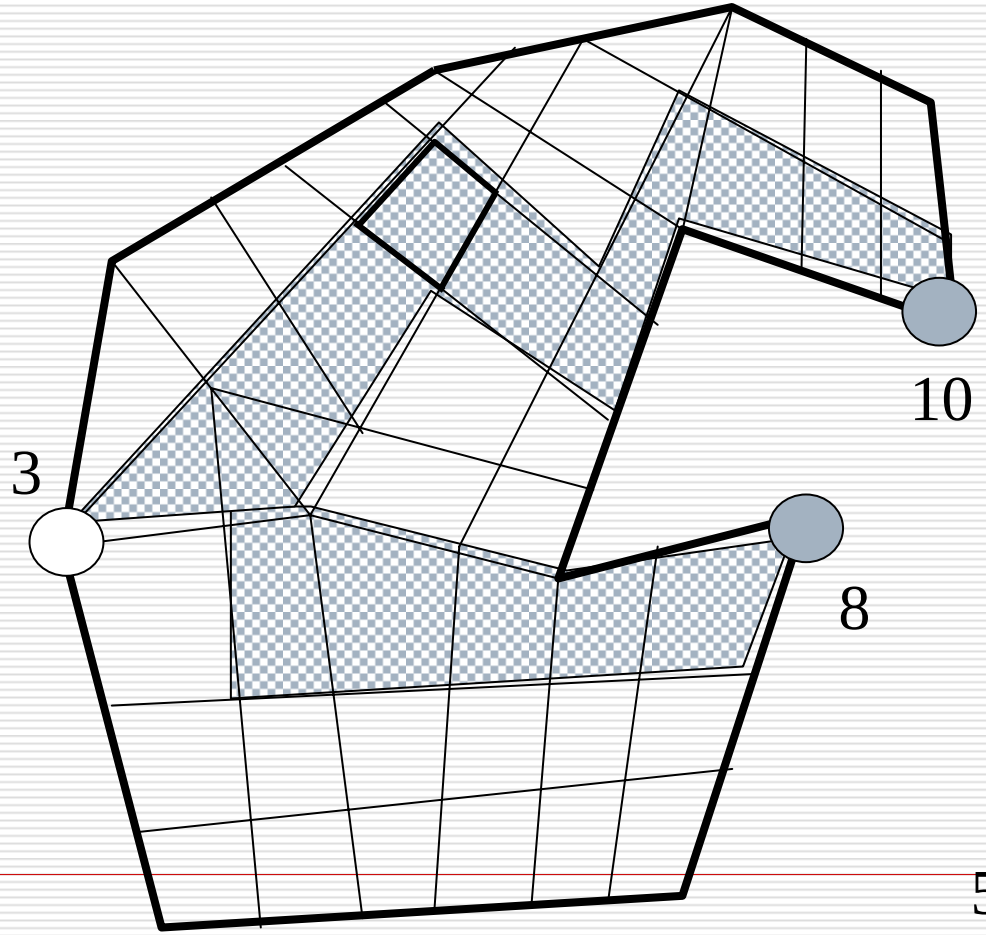


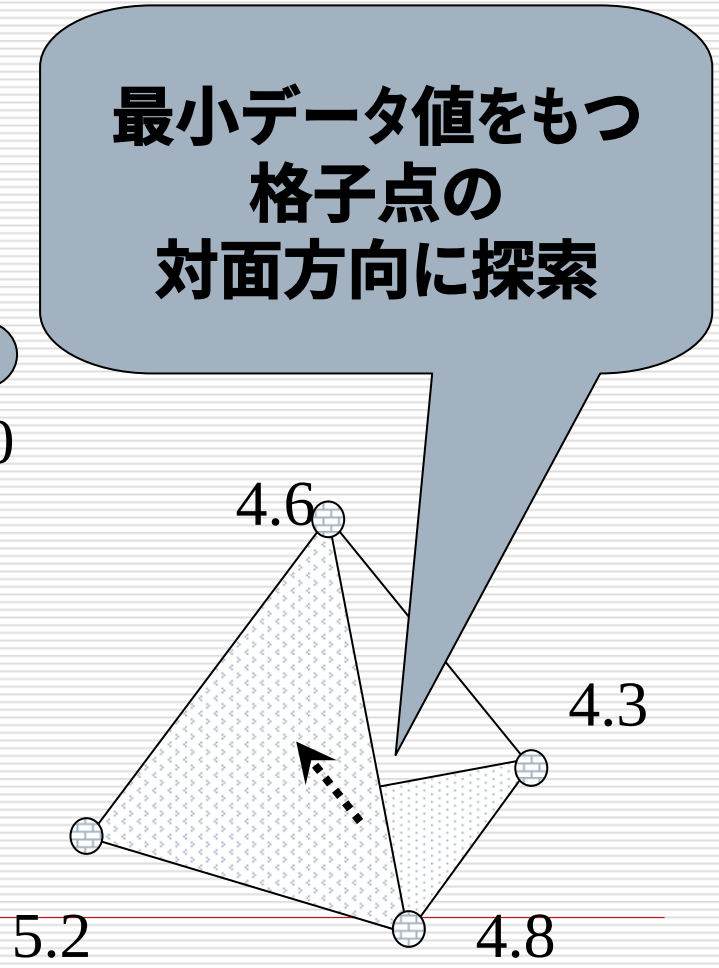
等値面と交差する格子
から
自己増殖的に等値面
を生成する

シード格子

極点からのシード格子探索

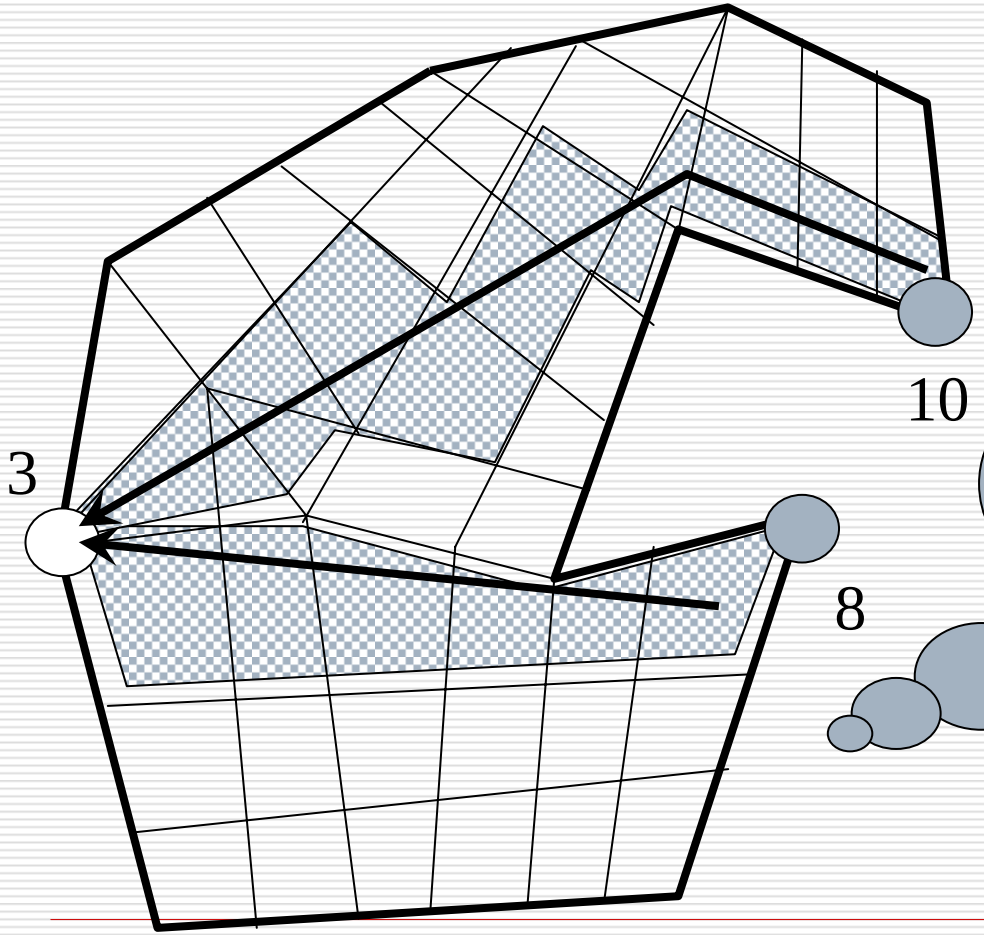
(Koyamada and Itoh, 1995)





極点グラフの生成

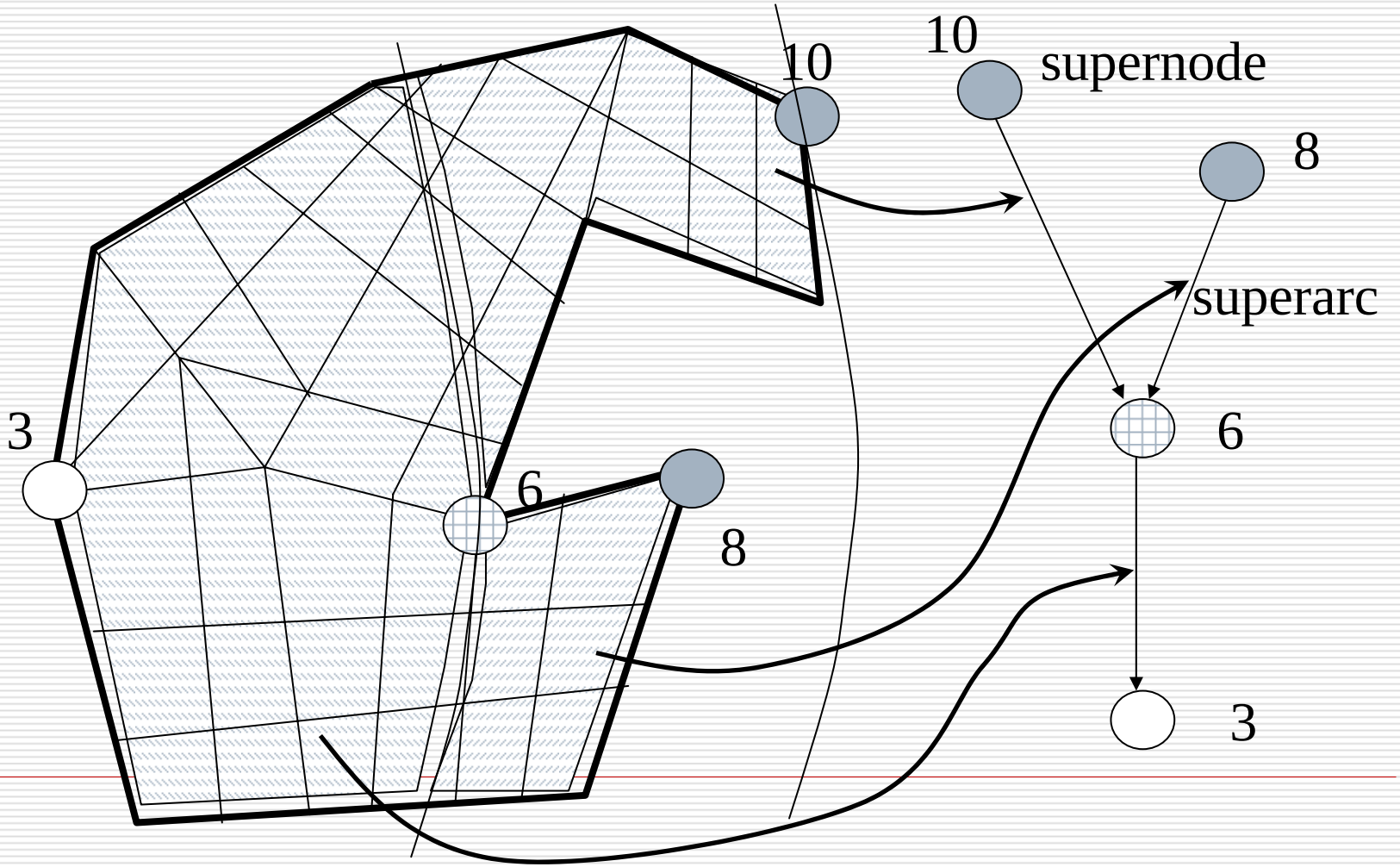
(Itoh and Koyamada, 1996)



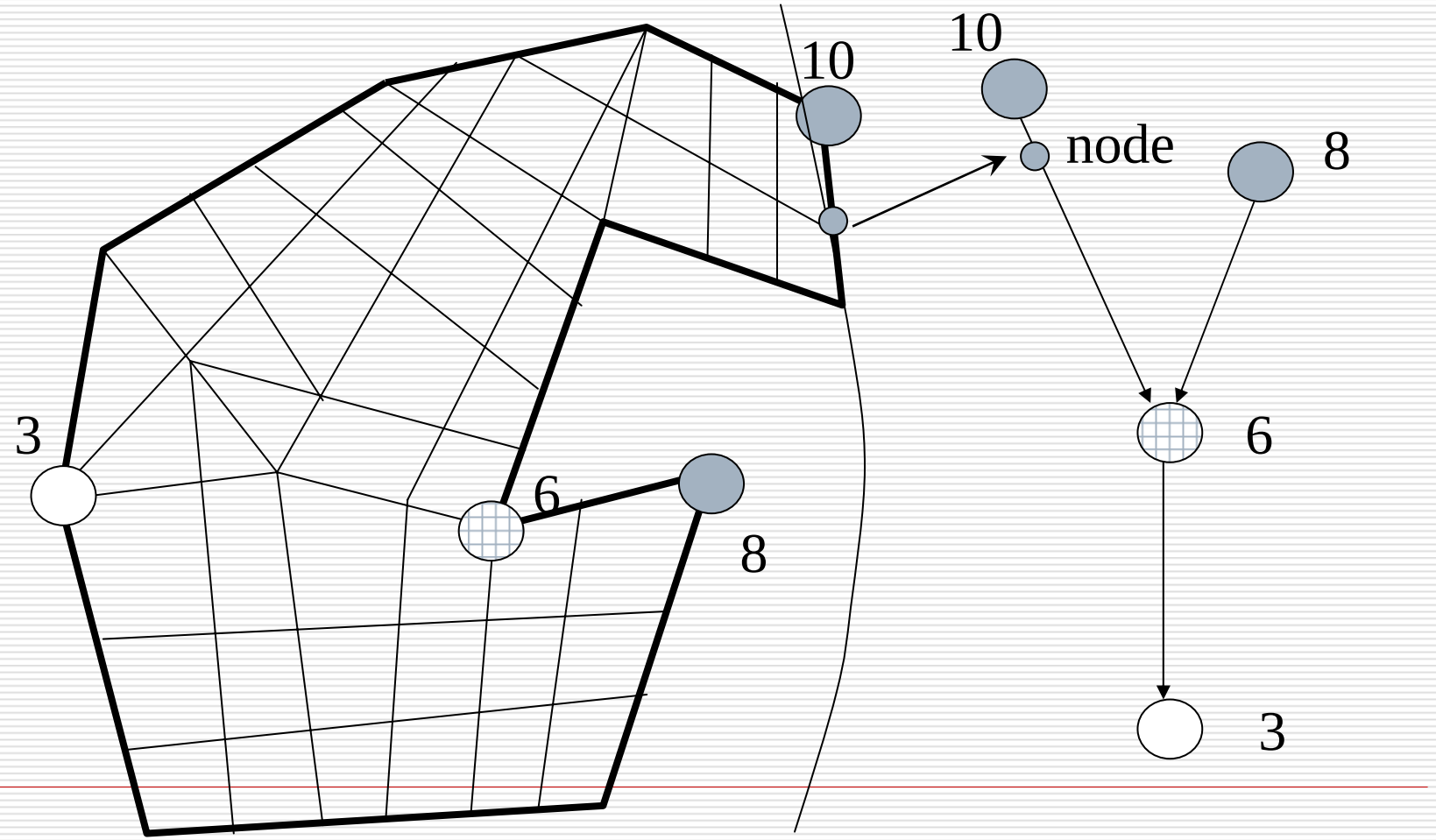
鞍点を考慮していない
ので貫通穴のある場合に
抜けがでてしまう

Contour Tree

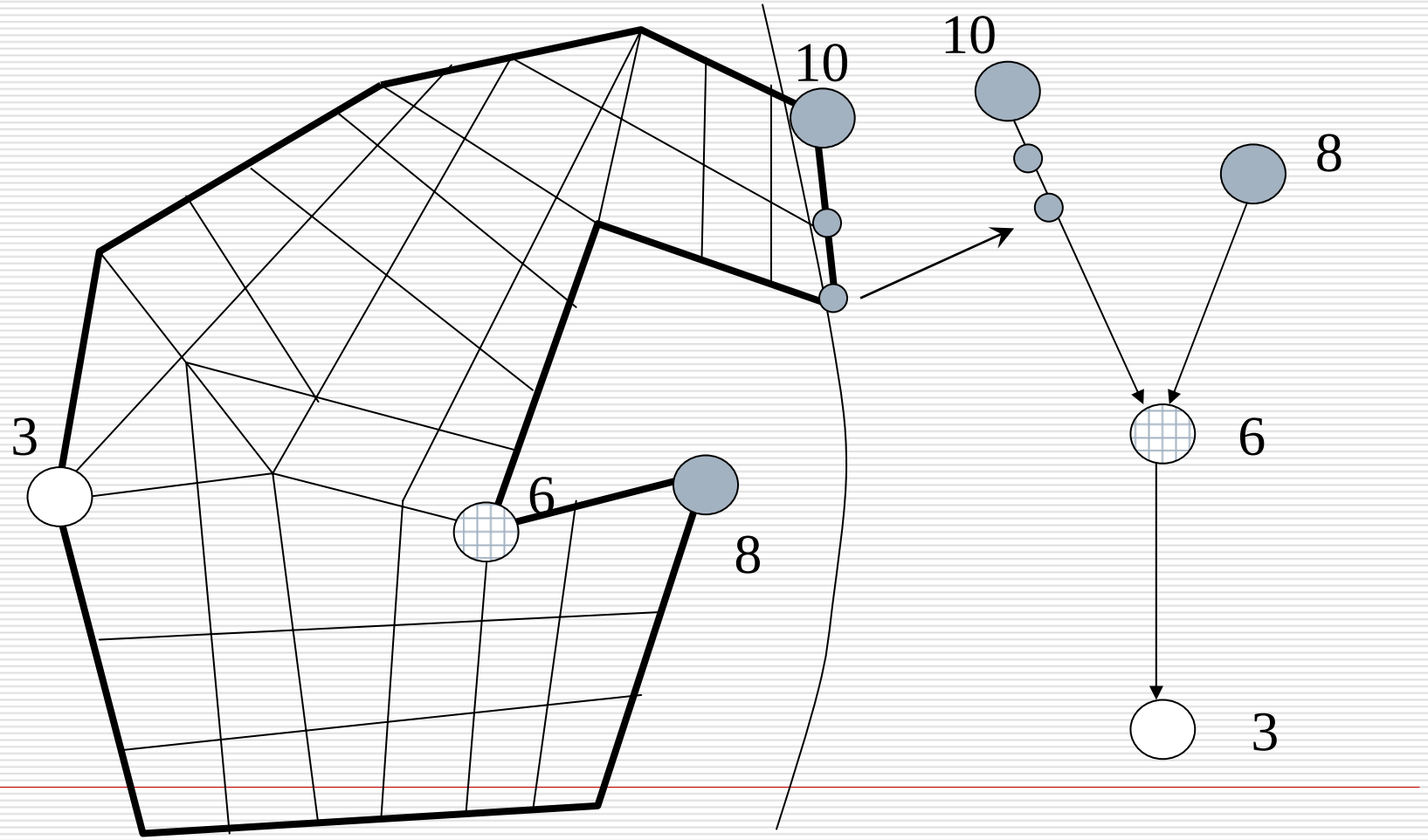
(Freeman and Morse, 1967)



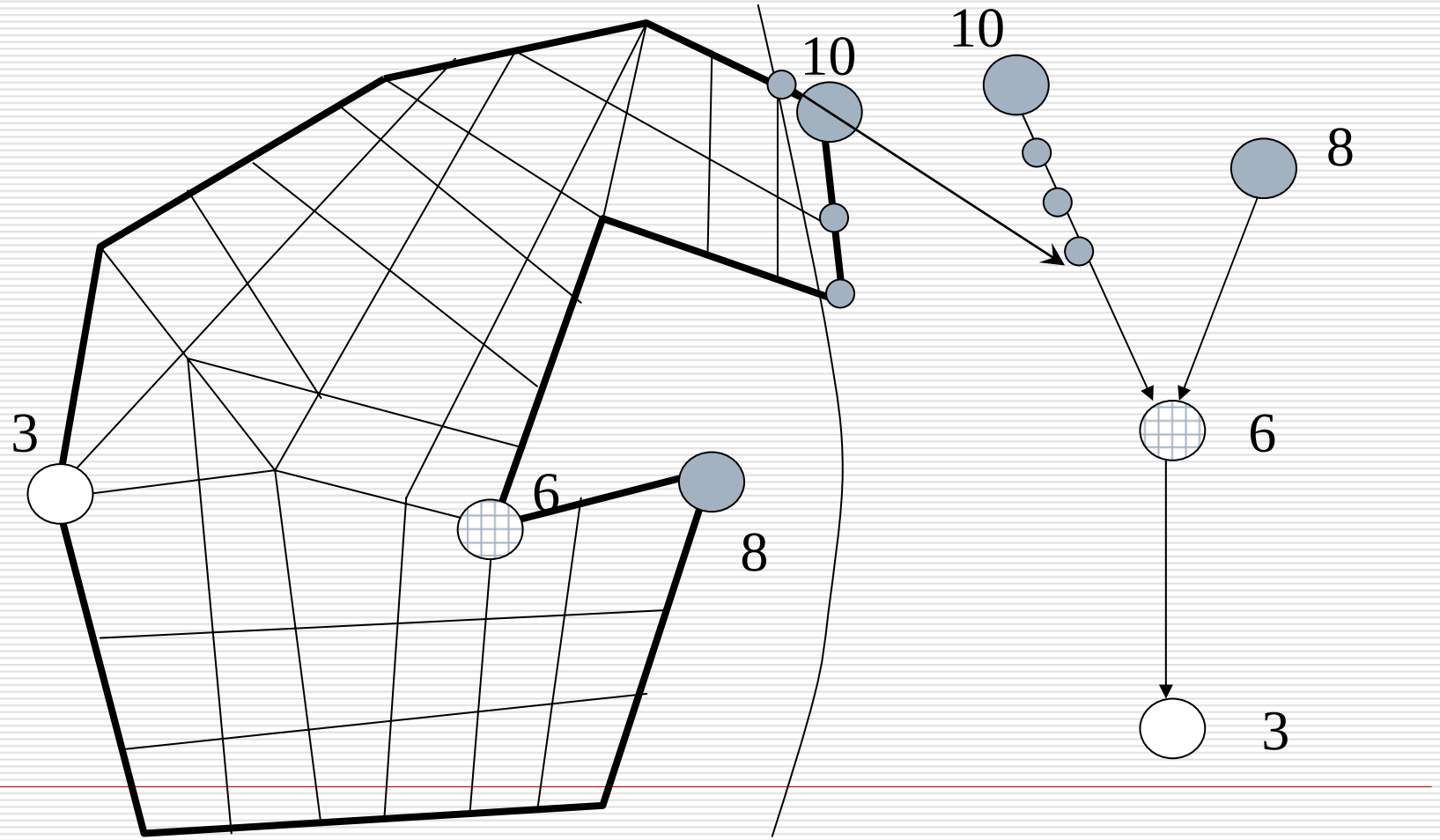
Contour Tree



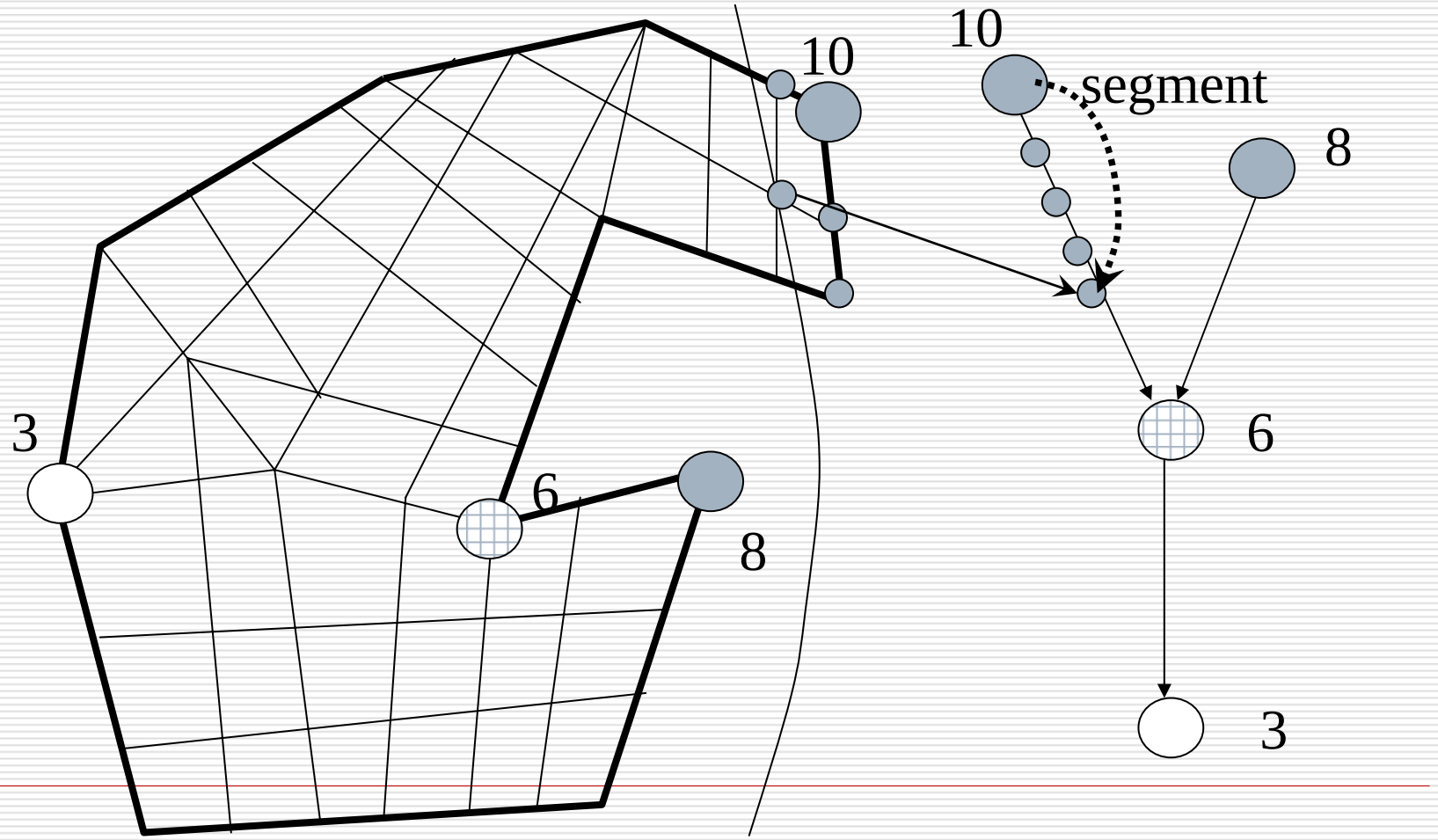
Contour Tree



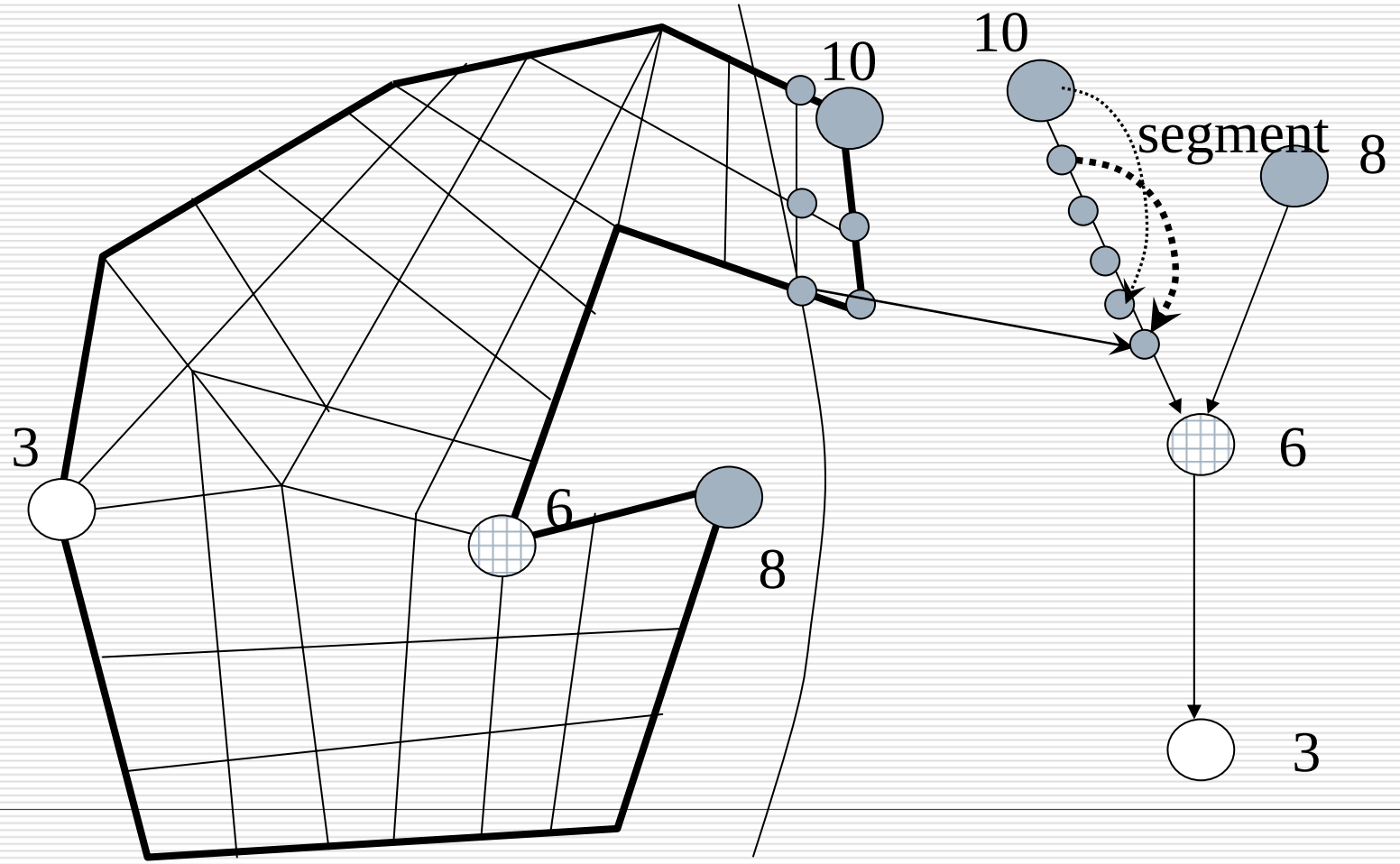
Contour Tree



Contour Tree

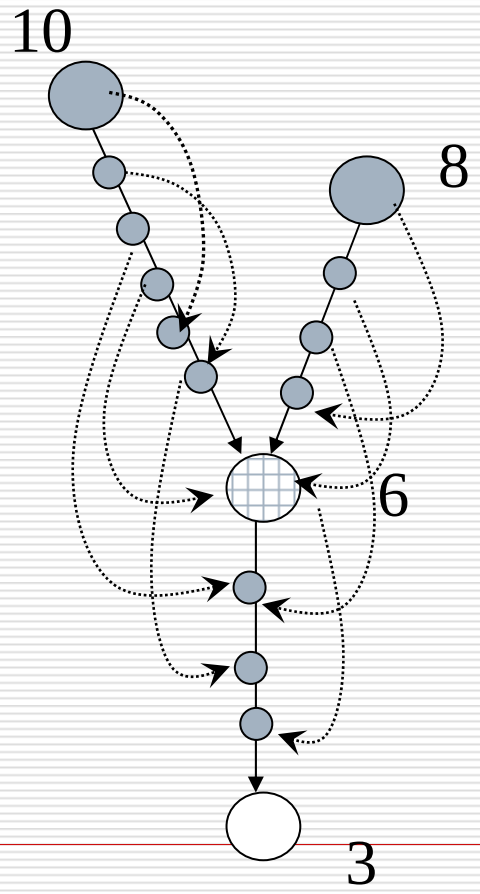


Contour Tree



Seed set selection

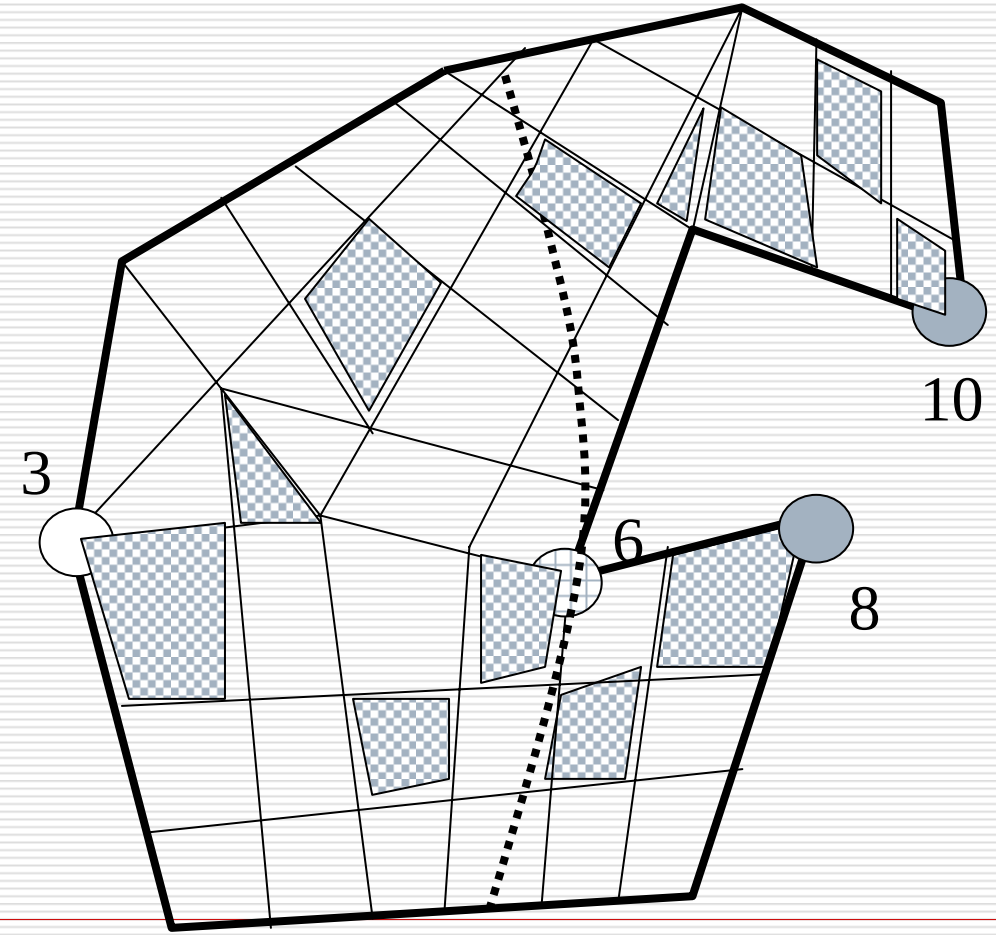
(Kreveld and et. al, 1994)



Find a small subset of the segments such that each arc of the tree is passed by some segment of the subset.

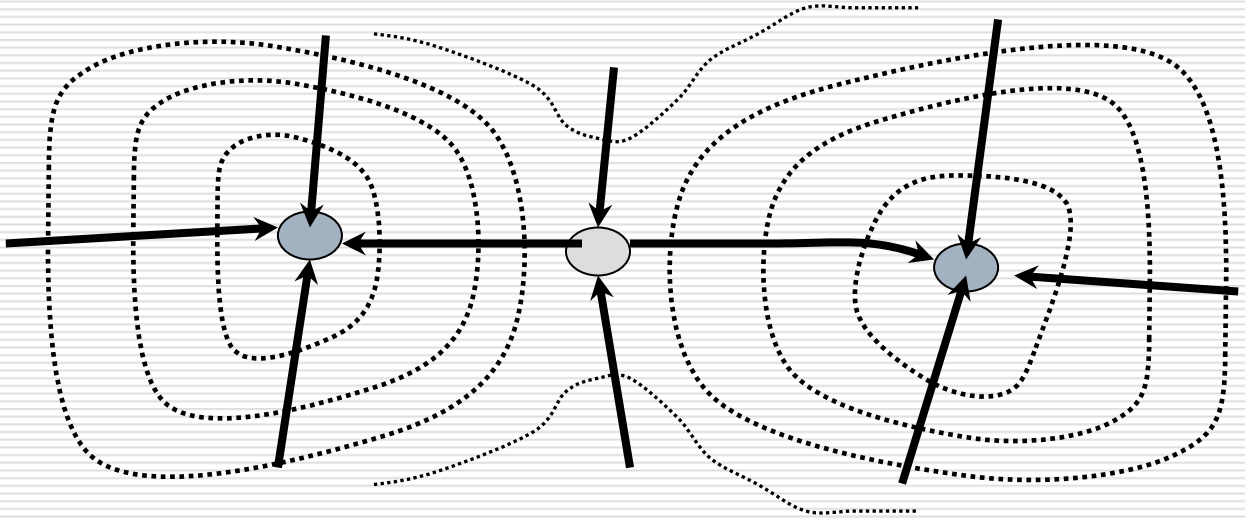
Simple greedy method
(Time complexity: $O(n^2)$)

シード格子選択結果例



Gradient flow topology
(C.L.Bajaj,1997)

- **スカラ場の勾配ベクタに特異点理論を適用**
  - **極大点、極小点、鞍点の探索**
  - **上記特異点をベクタ線で接続する**

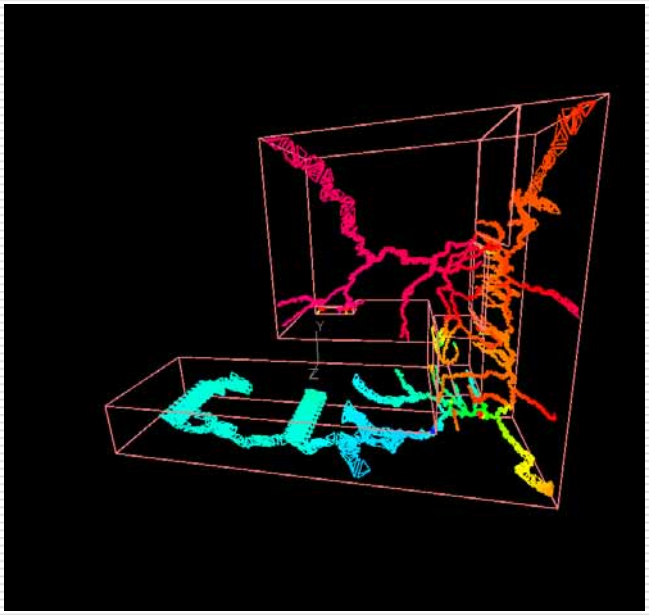


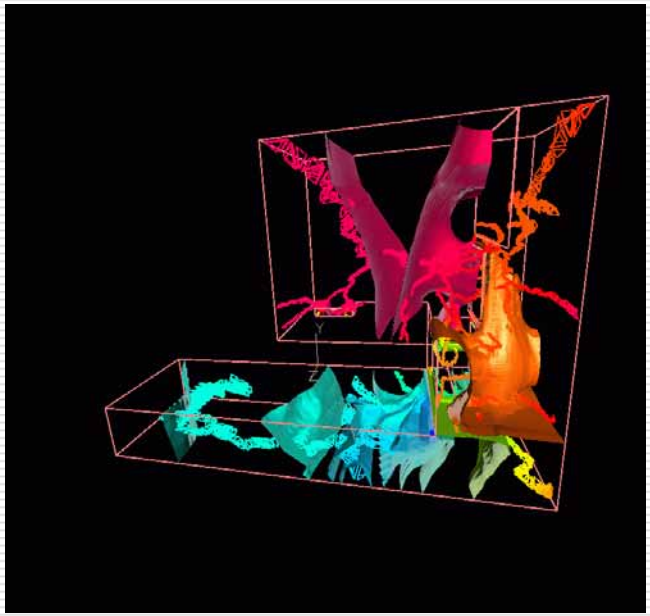
Volume thinning
(Itoh Yamaguchi and Koyamada, 1996)

- 画像処理における細線化技術の3D化
  - 極点を頂点として持つ格子は除去しない
  - その格子に接続する格子につながりが存在するなら除去する
  - そうでないなら除去する
- トポロジー情報の保持された極点グラフを効率よく作成(Time complexity: O(n))

ボリューム細線化例

- 細線化結果（左）と生成した等値面（右）

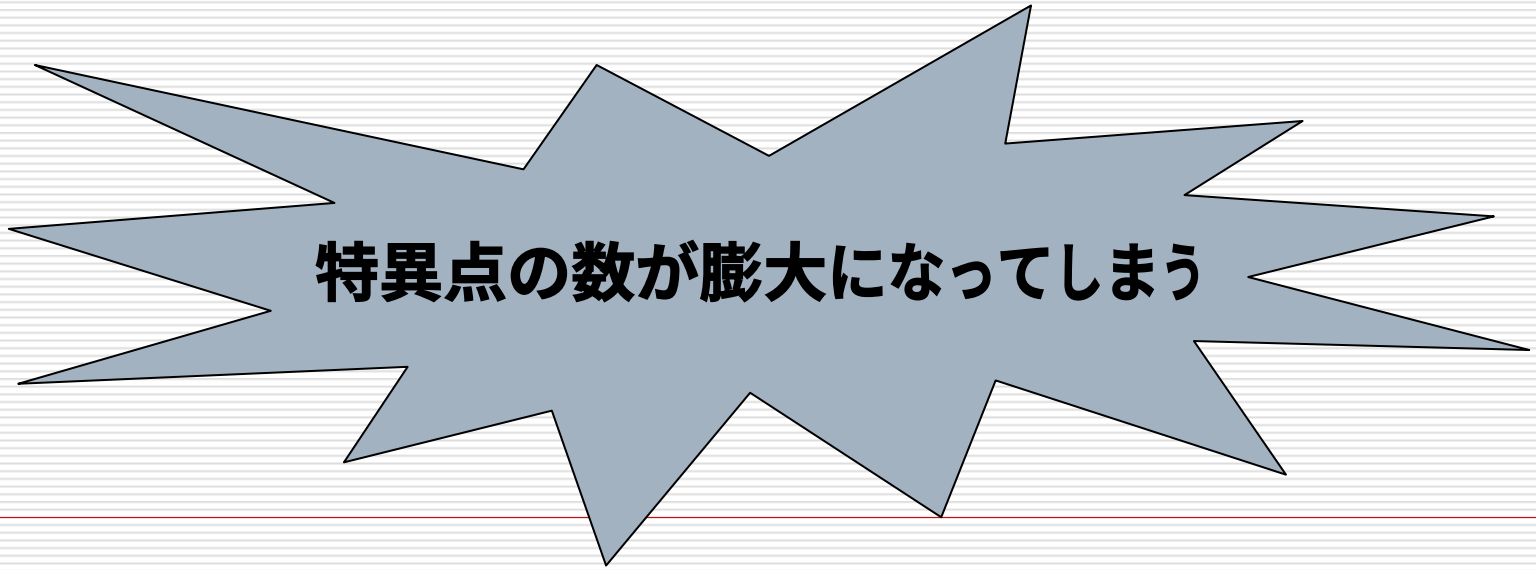




出所：http://www.viz.media.kyoto-u.ac.jp/

特異点ベースの手法の問題点

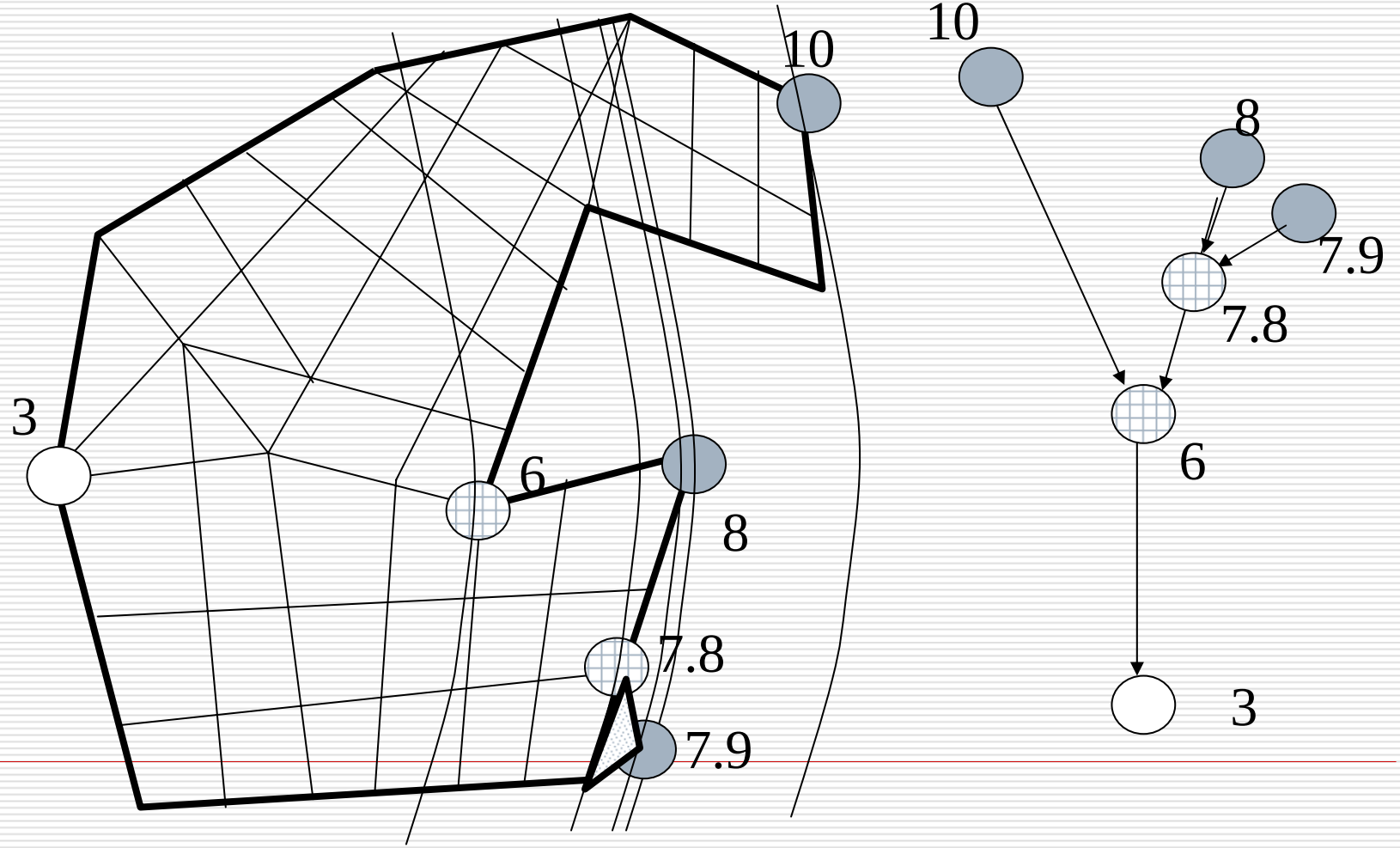
- ノイズに弱い
  - データ計測時に混入する場合
  - 計算が十分に収束していない場合。。



Digital Morse Theory
(Karron and Cox, 2000)

- To eliminate small noisy features,
  - label each point according to the critical point that owns it
  - construct a segmentation of the data by critical points
  - measure the relative importance of the criticality by the size of the basin of attraction

Example of a noisy feature



まとめ

- 特異点理論はボリュームビジュアライゼーションにとって有益
  - シード格子の特定
  - 膨大な情報の抽象化
- 特異点の影響範囲を調べることが重要
  - ノイズデータの見極め
  - 重要度の計算

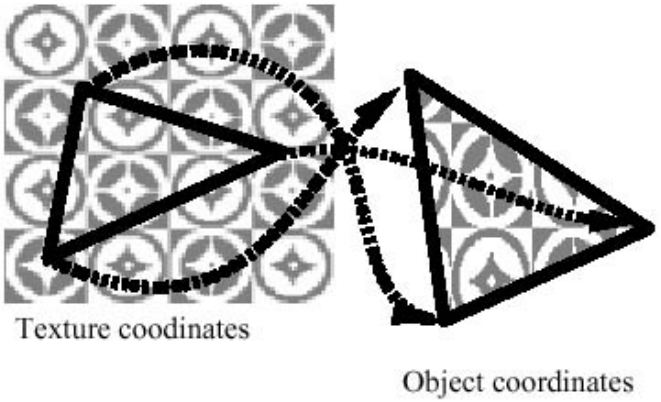
OpenGL ~プログラミング概論~

- 6 テクスチャマッピング
  - オブジェクトの外観をよくする手法の一つ
  - サーフェスの色を変えて“複雑な”サーフェスを表現するのに使うのが典型例
    - 木目調のテクスチャから平面ポリゴンが“木”に見えるようにする
  - ビジュアルリアリティとクオリティの向上
    - 環境マッピングを用いて周りの風景がオブジェクトに映り込むのを表現

OpenGL ~プログラミング概論~

- 6 テクスチャマッピング
- 6.1 テクスチャ座標
  - テクスチャ画像の位置を頂点に割り当てる
  - ラスタライズとテクスチャ座標

1. 頂点に割り当てられたテクスチャ座標はフラグメントのラスタライズ時に補間
2. フラグメントのテクスチャ座標をテクスチャの1つもしくは複数のテクセルの位置に変換
3. テクセルを取得し、色をフィルタリングし、フラグメントのテクスチャ色に
4. フラグメントのテクスチャ色とフラグメント色を合成




```
glEnable(GL_TEXTURE_2D);
glBegin( GL_TRIANGLES );
 for( i = 0; i < 4; i++ ){
  glNormal3dv( normal[ i ] );
  glTexCoord2f(-1.0, -1.0);
  glVertex3dv( tri_vertex[ triangle[ i ][ 0 ] ] );
  glTexCoord2f(1.0, -1.0);
  glVertex3dv( tri_vertex[ triangle[ i ][ 1 ] ] );
  glTexCoord2f(1.0, 1.0);
  glVertex3dv( tri_vertex[ triangle[ i ][ 2 ] ] );
 }
 glEnd();
```

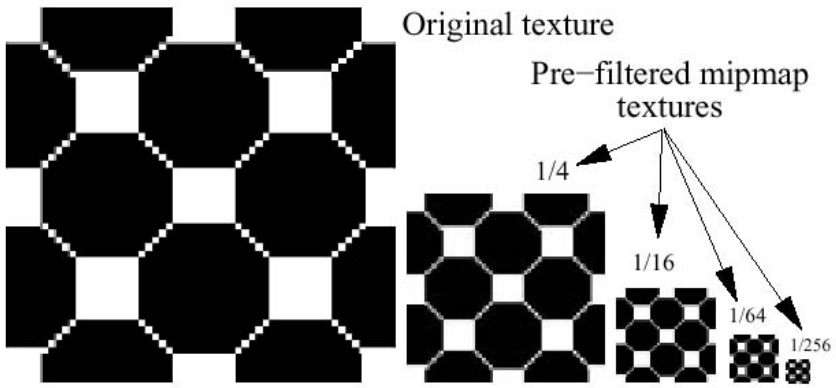
画像出所：http://www.bluevoid.com/opengl/sig00/advanced00/notes/node53.html

OpenGL ~プログラミング概論~

- 6 テクスチャマッピング
- 6.2 ミップマッピング(1)
  - テクスチャは複数のLevel-of-Detail(LODs)から成る
  - 各レベルは異なったテクスチャ画像
  - ベースのミップマップレベル(レベル0)がもっとも解像度が高い
  - レベルが増えるにつれてサイズは半分づつに減って行き、最後は1x1

OpenGL ~プログラミング概論~

- 6 テクスチャマッピング
- 6.2 ミップマッピング(2)
  - サンプリングは一つ前のレベルのダウンサンプリング
  - `gluBuild{1,2,3}Dmipmaps`
    - GLUのミップマップ生成関数




```
if (name) {
    free(image);
    image = glmReadPPM(name, &iwidth, &iheight);
    if (!image)
        image = glmReadPPM("data/fishermen.ppm",
&iwidth, &iheight);
 }
 gluBuild2DMipmaps(GL_TEXTURE_2D, 3, iwidth,
iheight, GL_RGB,   GL_UNSIGNED_BYTE, image);
```

出所：http://www.bluevoid.com/opengl/sig00/advanced00/notes/node57.html