

バイオメディカルエンジニアリング 5,6コマ目 画像処理概論(1)

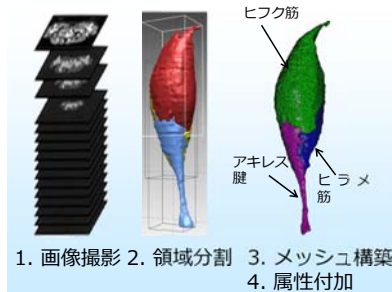
北大 & 理研 連携講座
2015/07/28

井尻敬
理研(客員研究員), 北大(客員准教授),
立命館(講師), 国循(客員研究員)

専門 : CG / UI / Image Processing

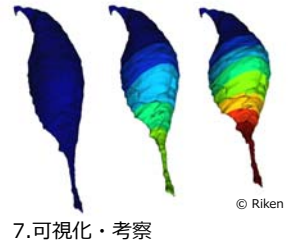
生体シミュレーションの流れ

モデリング



シミュレーション

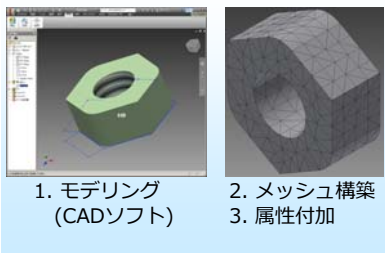
5. 計算モデル構築
6. 数値計算



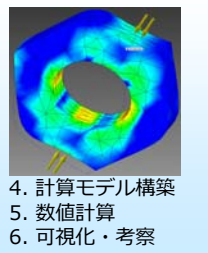
© Riken

CAD・CAE・CAMの流れ

モデリング (CAD)



シミュレーション(CAE)



製造・加工へ(CAM)

- ※ CAD : Design
- ※ CAE : Engineering
- ※ CAM : Manufacturing

生体シミュレーションとの大きな違いは
モデルの設計図があるところ

Contents - 画像領域分割 -

デジタル画像データ

- Vector Graphics と Raster Graphics
- 標準化と量子化
- 階調数とヒストグラム
- ダイナミックレンジ とHDR1

画像領域分割法

- 画像領域分割とは
- 閾値法
- 領域成長法
- クラスタリング
- 識別器
- 動的輪郭モデル
- グラフカット法
- 陰関数曲面再構成法

全部やると時間が無いので
面白いところだけ話します

Vector Graphics と Raster Graphics

Vector Graphics

画像を数式(スプライン等)で表現
計算機で描いたイラスト
例 wmf/ai/cdr/cgm/dfx等

Raster Graphics

画像をグリッド状の画素で表現
写真/CT/MRI等の観察画像
例 jpg/png/bmp/gif/tif/等

両者を含む : pdf/DjVu/eps/pict/ps/swf/xaml等



Vector Graphics と Raster Graphics

Vector Graphics

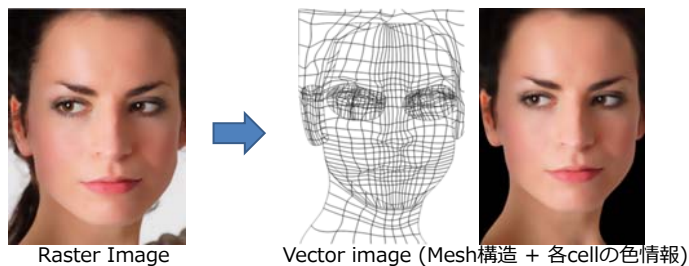
制御点のみを保持するため
データが小さい
拡大しても輪郭がスムーズ
計算機で描いたイラストに向く

Raster Graphics

画素情報を保持するため
データが大きい
拡大したらギザギザ
風景など自然の画像に向く
→ 画像領域分割で扱うのはこちら



Raster → Vector 変換 (Vectorization)



Gradient mesh (Adobe Photoshop)

課題 : Raster image (写真等) を Vector image に変換したい
方法 : 画像の特徴線に沿うメッシュを構築し頂点に色情報を保持
各パッチ (四角形) で頂点の色を滑らかに混ぜる

図は[Jian Sun et. al. Image vectorization using optimized gradient meshes, TOG 2007]より

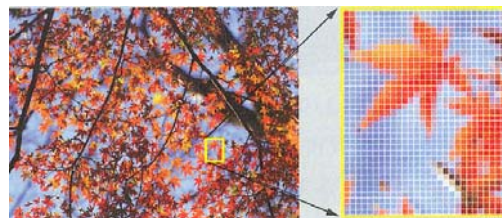
標本化と量子化

デジタル画像とは『離散値 (画素) が格子状に並んだデータ』
アナログ情報からデジタル画像を取得するとき

標本化: 空間の離散化

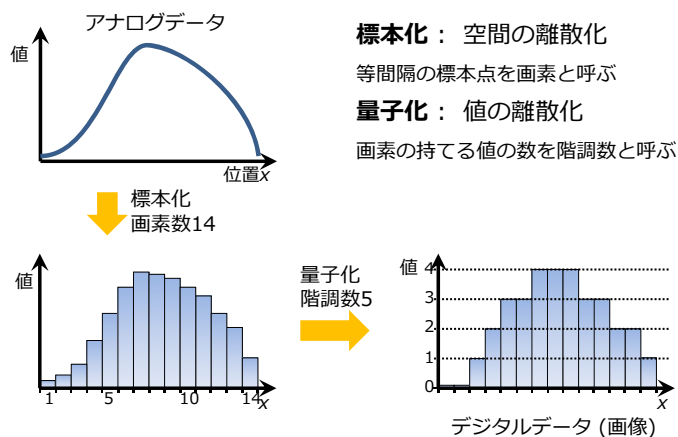
量子化: 値の離散化

の必要がある



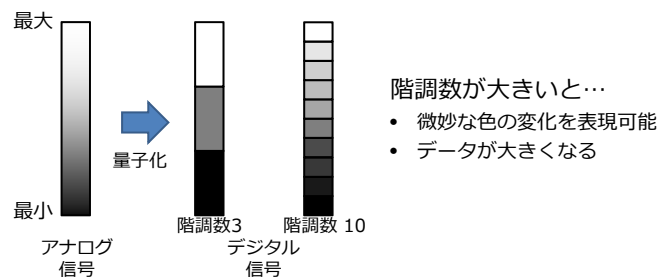
© CG-ARTS境界 デジタル画像処理

標本化と量子化



『階調数』 (又は、画素深度・色深度) とは

デジタル画像の各画素の色数のこと
最小値と最大値の分割数



画像フォーマットの階調数

ビットマップ(.bmp)

1bit bitmap : モノクロ画像
4/8bit bitmap : 16/256色のカラーパレット(インデックスカラー)
16/24bit bitmap : RGB毎に 5/8-bit 階調

Portable Network Graphics (.png)

グレースケール : 1, 2, 4, 8, 16-bit階調
カラー : 24bit (RGB毎に8bitの階調数), 48bit
インデックスカラー : 1, 2, 4, 8個のカラーパレット

Nikon D7000(rawデータ)

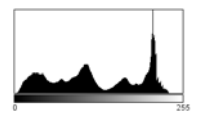
14bit

某社 X線マイクロCTの生データ (rawデータ)

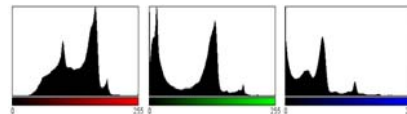
12bit階調 (階調数 = 濃度分解能)

『頻度表 (ヒストグラム)』とは

各階調の画素数を数えた表のこと
回転や平行移動に依存しない特徴量 → 画像処理に頻出



グレースケール画像



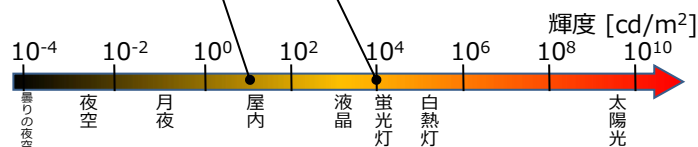
RGBカラー画像

ここから少し脱線して
「High dynamic range imageとトーンマッピング」
の話をします

『輝度(Luminance)』とは



人の感じる物体の明るさ
ある光源に対して
- 単位方位角あたり
- 見かけの単位面積あたり
『人の感じる』明るさ



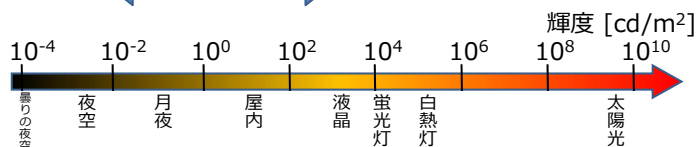
『ダイナミックレンジ』とは

信号をセンサーで計測するとき
計測可能な最小輝度値 I_{min} と最大輝度値 I_{max} の幅のこと

$$D = 20 \log_{10} \frac{I_{max}}{I_{min}} \text{ (db)}$$

デジタルカメラ用ccdセンサ
80db - 120db

人の視覚のダイナミックレンジは
• ある視野内で100db程度
• 順応を考慮すると200db以上
[奥田: 高ダイナミックレンジ画像. 2010]



HDRI : Motivation



露光時間 1/500 秒
黒つぶれ (アンダー)

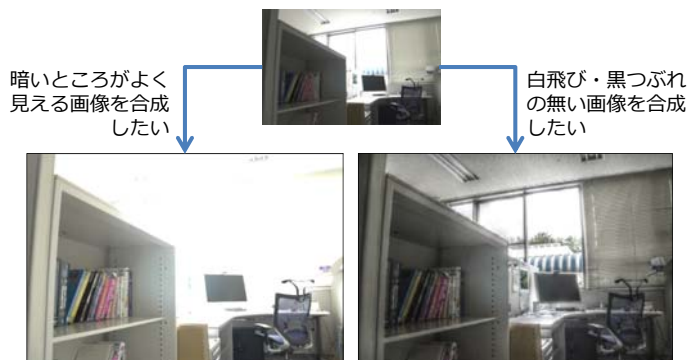


露光時間 1/4 秒
白飛び (オーバー)

デジタルカメラのセンサは
『凄く明るいところ』と『凄く暗いところ』
を同時に撮影できない

HDRI : Motivation

『凄く明るいところ』と『凄く暗いところ』の情報を持つ画像
を取得して…



HDRI : HDRIとトーンマッピング

HDRI



黒つぶれ・白飛びがなく
大きな/小さな値をもつ画像
階調数(RGB毎): 16bit
輝度値の幅: $[10^{-1}, 10^4]$



表示のため階調数を落とす処理が必要
『トーンマッピング』

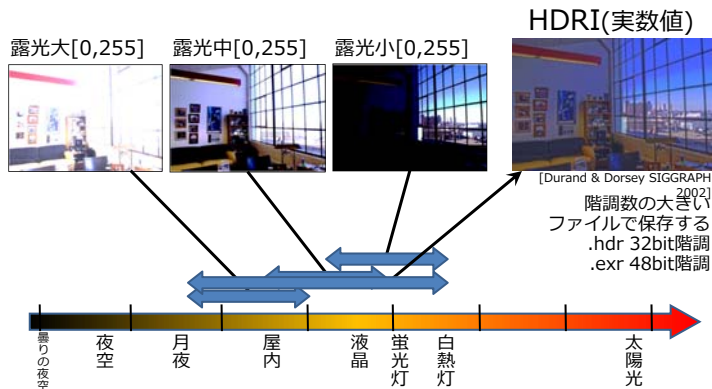
**表示
デバイス**



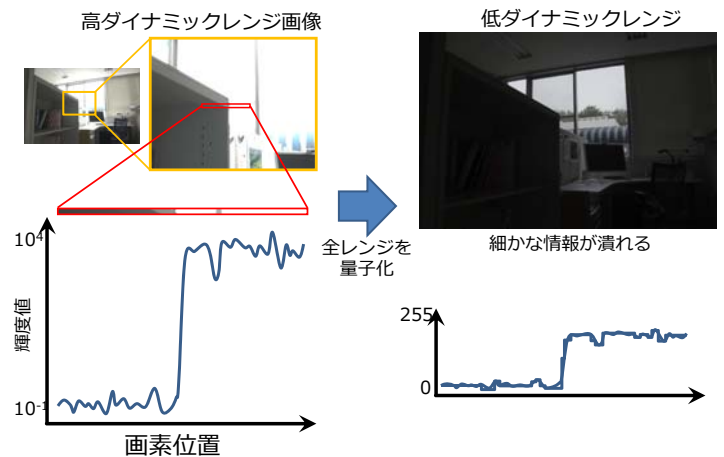
液晶モニター・プロジェクタ等
階調数(RGB毎): 8bit $[0, 255]$

HDRIの取得

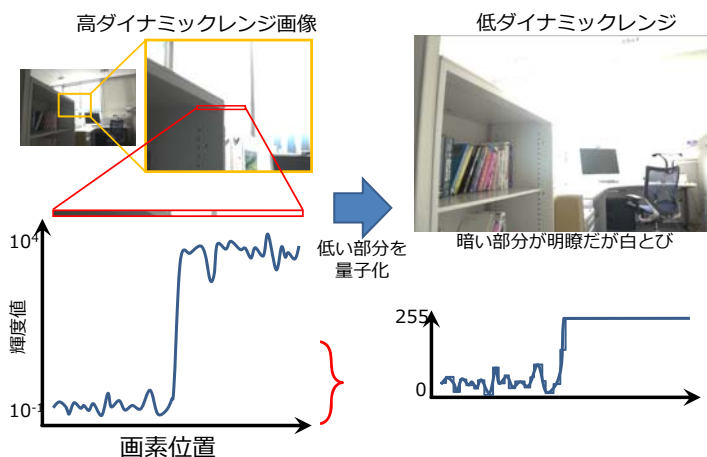
最小輝度値と最大輝度値の幅の広い画像
露光を変えた写真の撮影から合成できる



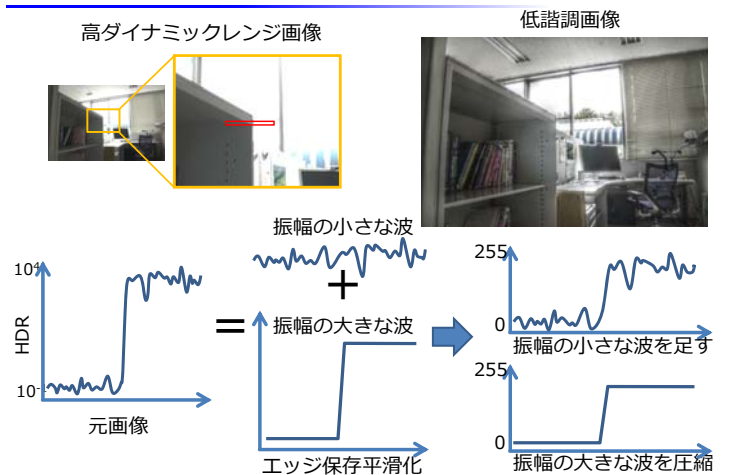
トーンマッピング (線形)



トーンマッピング (線形)



トーンマッピング (HDR合成)

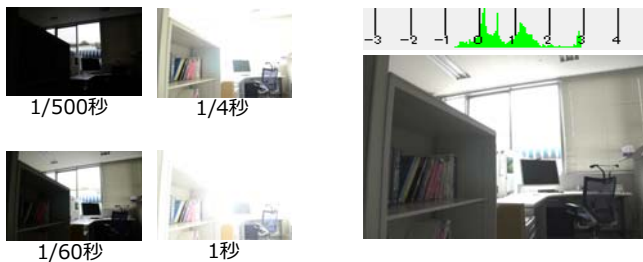


HDRIを自作してみる

Luminance HDR

露光時間が可変のカメラ - NIKON D7000
HDRI合成ソフト - Luminance HDR (ver 2.3.1)

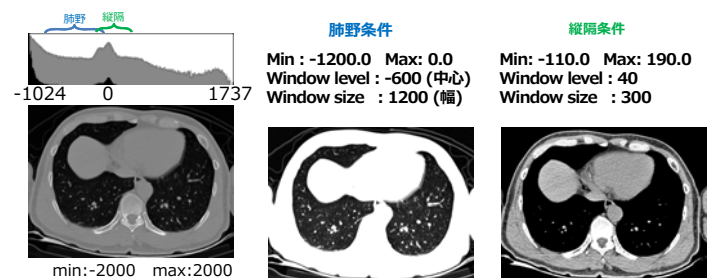
1. 露光時間を変え撮影 2. 一枚のHDRI画像に合成



CT画像のトーンマッピング

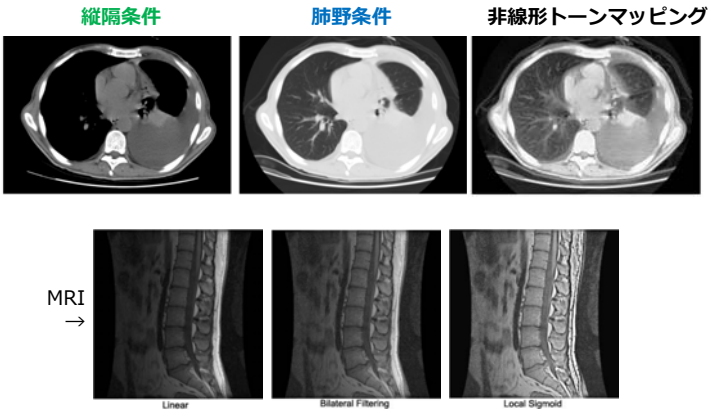
CT画像

階調数 : 12 - 16 [bit]
レンジ : -1000 - 1500 [HU]
→ トーンマッピングの必要有

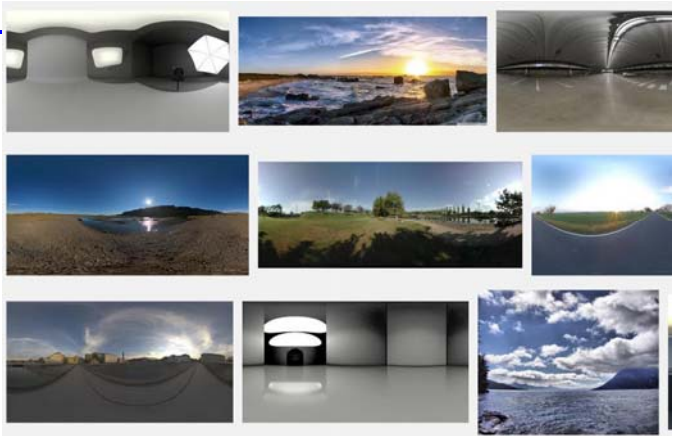


画像は理化学研究所生体力学シミュレーションチームより

CT / MRI 画像のトーンマッピング



上図: F. Edward Boas, "High dynamic range images in radiology 2007" www.stanford.edu/~boas/science/pub_list.html
下図: Park et al. "Evaluating Tone Mapping Algorithms for Rendering Non-Pictorial (Scientific) High-Dynamic-Range Images", JVCI 2007.



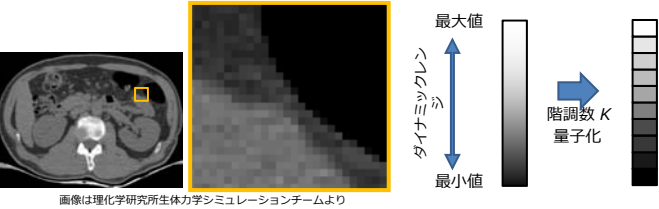
図はHDRでGoogle画像検索

まとめ 『デジタル画像データ』

『Vector Graphics』『Raster Graphics』『階調数』『ヒストグラム』『ダイナミックレンジ』『高ダイナミックレンジ画像(HDRI)』『トーンマッピング』

領域分割の対象：Raster画像

階調数が8bitの画像データ (e.g. bmp, png) は、rawデータが量子化されている(情報がつぶれている)可能性がある



Contents -画像領域分割-

デジタル画像データ

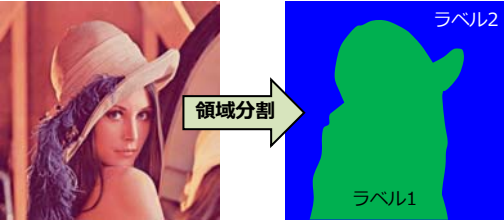
- Vector Graphics と Raster Graphics
- 標準化と量子化
- 階調数とヒストグラム
- ダイナミックレンジ とHDRI

画像領域分割法

- 画像領域分割とは
- 閾値法
- 領域成長法
- クラスタリング
- 識別器
- 動的輪郭モデル
- グラフカット法
- 陰関数曲面再構成法

画像領域分割 (Image Segmentation) とは

- 『画像領域分割』『画像領域抽出』『画像ラベリング』
- Vision/Graphics/Image Processing 分野において重要な研究分野
- デジタル画像の各画素にラベルをつける作業 (ラベル画像を作る作業)



Low-level と High-level segmentation

Low-level segmentation

画像を特徴 (色等) が一般的な局所領域に分割する作業

High-level segmentation

画像内の目標物の領域を切り抜く作業

※両者の境界は曖昧で両者の意味を込めて『画像領域分割』と呼ぶのが一般的



例：Water shed法



例：Graph Cut法

Low-level と High-level segmentation

Low-level segmentation

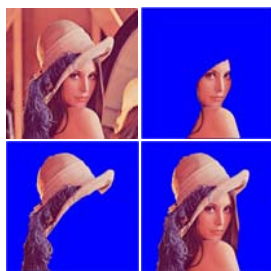
意味のある固まりを抽出
画像を圧縮
処理の高速化
(画素は直接処理するのに小さすぎる)

High-level segmentation

画像編集 (エフェクト適用)
コラージュ
シミュレーション用モデルの構築(3D)



例: Water shed法

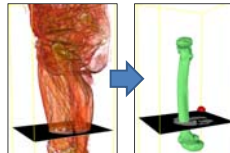


例: Graph Cut法

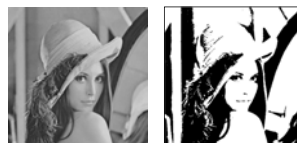
二値化と多値化

二値化

画像を前景・背景に分割

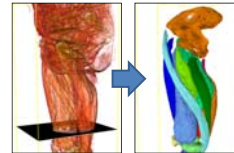


例)濃淡画像の白黒二値化

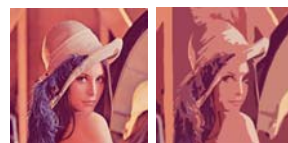


多値化

画像を複数領域に分割



例) ポスタリゼーション
(階調数を削減し特殊効果を得る)



上図のCTデータは理研より

画像領域分割法の計算法による分類

- | | |
|------------------------------------|------------------------------|
| - 閾値法 (Thresholding) | : Tthresholding.exe |
| - 領域成長法 (Region Growing) | : TRegionGrowing.exe |
| - クラスタリング (Clustering) | : TClustering.exe |
| - クラス分類器 (Classifier) | : ---- |
| - 動的輪郭モデル(Active Contours) | : TActiveContour.exe |
| - グラフカット法 (Graph Cut Segmentation) | : TGraphCut.exe |
| - 曲面再構成法 (Surface Reconstruction) | : TSurfaceReconstruction.exe |

※ 上記の各手法が独自の研究分野を築くほど広がりを持つ

例) Otsu et. al. 1975. ヒストグラムを用いた閾値法 の引用数 12643

例) Kass et. al. 1988. Active Contourの最初の論文の引用数 14843

例) Boykov et. al. 2001. Graph Cut法の初期の論文の引用数 3764

※ Google scholar (2013年7月時点)

※ 上記の分類間をまたぐ手法や上記の分類に入らない手法も存在

Contents -画像領域分割-

デジタル画像データ

- Vector Graphics と Raster Graphics
- 標本化と量子化
- 階調数とヒストグラム
- ダイナミックレンジ とHDRI

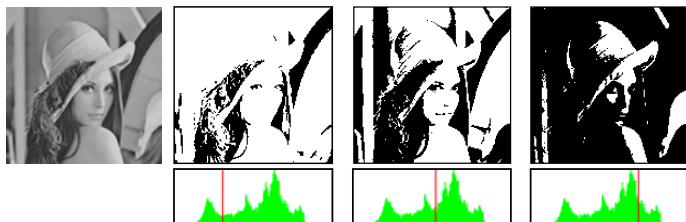
画像領域分割法

- 画像領域分割とは
- 閾値法
- 領域成長法
- クラスタリング
- 識別器
- 動的輪郭モデル
- グラフカット法
- 陰関数曲面再構成法

閾値法 Thresholding

閾値法 > 概要

TThresholding.exe



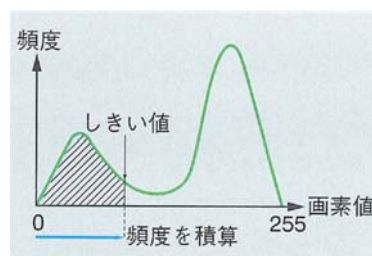
閾値により画素に前景・背景ラベルを付ける

閾値を自動的に計算する方法が研究される

→ Pタイル法[1], 大津法[2], Sauvola法[3]...etc...

閾値法 > P-タイル法

領域の占める画素数(割合)を入力して画像を二値化する



入力: 黒領域の画素数 N

- ヒストグラムを構築
- 下から頻度を積算し、画素数が N に達した値を閾値に

[ビジュアル情報処理CG-ARTS境界]

[1] CG-Arts協会. ビジュアル情報処理-CG・画像処理入門- 2012

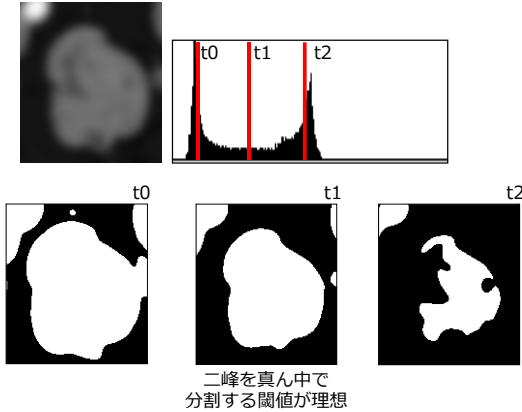
[2] Otsu N.: A threshold selection method from gray-level histo-grams. IEEE SMC, 9, 1979, 62-66.

[3] J. Sauvola et. al., "Adaptive document image binarization," Pattern Recognition 33(2), 225-236, 2000.

閾値法 > 大津法 (大切)

TThresholding.exe

二峰性のヒストグラムを持つ画像を仮定し
二峰を最も良く2分割する閾値を自動計算する

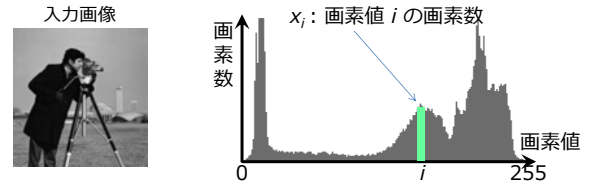


閾値法 > 大津法

TThresholding.exe

二峰を最も良く2分割したいので…

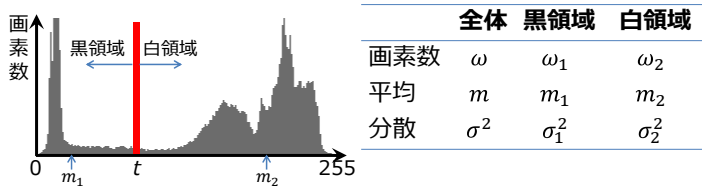
ヒストグラムの**分離度**を定義しこれを最大化する閾値を探す



$$\omega = \sum_{i=0}^{255} x_i \quad \text{平均} \quad m = \frac{1}{\omega} \sum_{i=0}^{255} x_i \times i \quad \text{分散} \quad \sigma^2 = \frac{1}{\omega} \sum_{i=0}^{255} x_i \times (i - m)^2$$

閾値法 > 大津法 > 分離度

ある閾値 t で2領域に分離したとき…



黒領域

$$m_1 = \frac{1}{\omega_1} \sum_{i=0}^{t-1} x_i \times i$$

$$\sigma_1^2 = \frac{1}{\omega_1} \sum_{i=0}^{t-1} x_i \times (i - m_1)^2$$

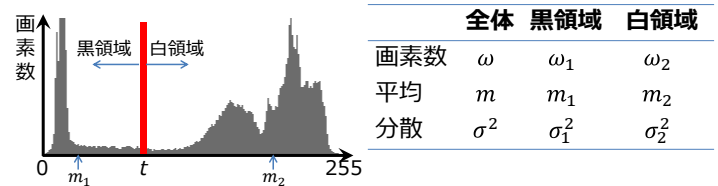
白領域

$$m_2 = \frac{1}{\omega_2} \sum_{i=t}^{255} x_i \times i$$

$$\sigma_2^2 = \frac{1}{\omega_2} \sum_{i=t}^{255} x_i \times (i - m_2)^2$$

閾値法 > 大津法 > 分離度

ある閾値 t で2領域に分離したとき…



クラス内分散

$$\sigma_w^2 = \frac{\omega_1 \sigma_1^2 + \omega_2 \sigma_2^2}{\omega_1 + \omega_2}$$

2領域の分散の和(正規化)
小さい方が良い分割

クラス間分散

$$\sigma_b^2 = \frac{\omega_1 (m_1 - m)^2 + \omega_2 (m_2 - m)^2}{\omega_1 + \omega_2}$$

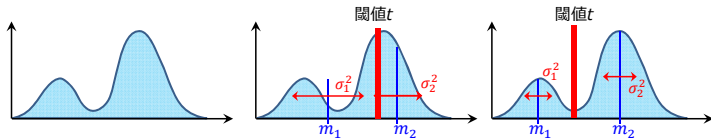
2領域の平均値の距離
大きい方が良い分割

閾値法 > 大津法 > 分離度

$$\text{分離度} = \frac{\text{クラス間分散}}{\text{クラス内分散}} = \frac{\sigma_b^2}{\sigma_w^2}$$

$$\sigma_b^2 = \frac{\omega_1 (m_1 - m)^2 + \omega_2 (m_2 - m)^2}{\omega_1 + \omega_2}$$

$$\sigma_w^2 = \frac{\omega_1 \sigma_1^2 + \omega_2 \sigma_2^2}{\omega_1 + \omega_2}$$

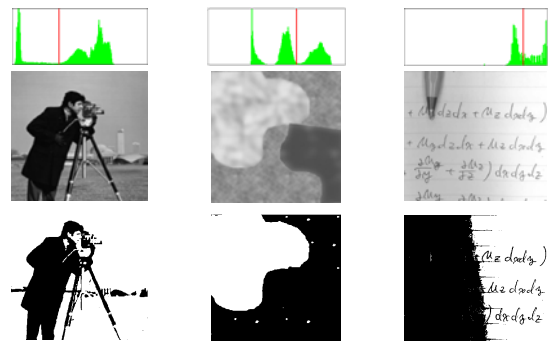


大津法

- 1) 入力画像のヒストグラムを構築
- 2) 閾値 t を 1 から 254 まで動かして**分離度**を計算
- 3) 分離度が最大になる閾値 t_{max} で画像を分割

閾値法 > 大津法の特徴

双峰性の高いヒストグラムを持つ画像には強い
(双峰性の低い画像には使えない)
グラデーションに弱い



閾値法 > Adaptive Thresholding TThresholding.exe

画素 (i,j)

各画素 x_{ij} 周囲の局所窓を考える
窓内のヒストグラムから**その画素用の閾値 t_{ij}** を計算

大津法：局所窓のヒストグラムから大津法を計算

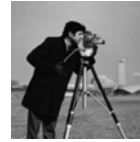
Sauvola法：局所窓の平均値と分散 σ^2 から閾値 t を計算

$$t_{i,j} = m_{i,j} \left(1 + k \left(\frac{\sigma_{i,j}}{R} - 1 \right) \right)$$

$t_{i,j}$: 画素 i,j の閾値
 $m_{i,j}$: 窓内の平均
 $\sigma_{i,j}^2$: 窓内の分散
 R : 最大標準偏差 (= 128)
 k : パラメータ (= 0.2)

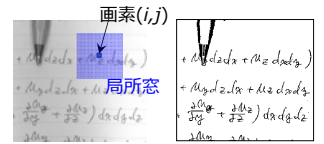
閾値法 > まとめ

大津法



$$\text{分離度} = \frac{\text{クラス間分散}}{\text{クラス内分散}} = \frac{\sigma_b^2}{\sigma_w^2}$$

Adaptive thresholding



局所窓の情報を利用して閾値計算
大津法 や Sauvola法

閾値により画素に前景ラベル・背景ラベルを付ける
閾値を自動計算する手法を紹介した
→ Pタイル法, 大津法, Sauvola法

Contents -画像領域分割-

デジタル画像データ

- Vector Graphics と Raster Graphics
- 標本化と量子化
- 階調数とヒストグラム
- ダイナミックレンジ とHDR1

画像領域分割法

- 画像領域分割とは
- 閾値法
- 領域成長法
- クラスタリング
- 識別器
- 動的輪郭モデル
- グラフカット法
- 陰関数曲面再構成法

クラスタリング Clustering

クラスタリング > 概要



RGB空間の画素の分布



12領域

全画素を**特徴空間**に射影し、特徴空間内で**密集する画素集合（クラスタ）**を同一領域として分割する

特徴空間 : 色空間, Bilateral空間, テクスチャ空間, etc...

有名な手法 : K-mean法[1], Mean shift法[2], Normalized Cut法[3], etc...

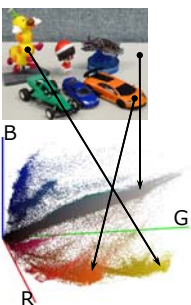
- [1] 高木幹雄ら, 新編画像解析ハンドブック. 東京大学出版会, 2004.
 [2] Comaniciu D. et. al.: Mean shift: A robust approach toward feature space analysis, *IEEE PAMI*, 24, 5(2002), 603-619.
 [3] Shi J. et. al.: Normalized cuts and image segmentation. *IEEE PAMI*, 22, 8(2002), 888-905.

クラスタリング > 特徴空間 TCluster.exe

特徴空間：画像の局所的な特徴が張る空間のこと

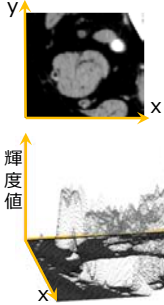
RGB空間

$$p_i \rightarrow (R_i \ G_i \ B_i)^T$$



Bilateral空間

$$p_i \rightarrow (p_{i,x} \ p_{i,y} \ I_i)^T$$



Bilateral空間 (Color)

$$p_i \rightarrow (p_{i,x} \ p_{i,y} \ R_i \ G_i \ B_i)^T$$

※ $R_i \cdot G_i \cdot B_i \cdot I_i$ は画素 p_i のR・G・B・輝度値

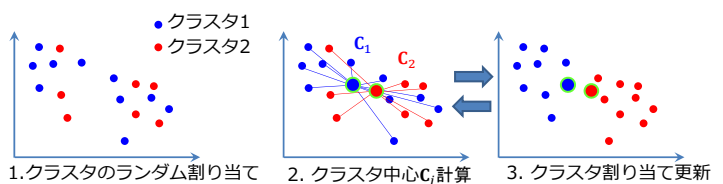
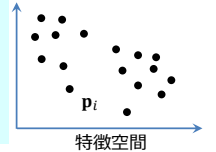
他にも...
テクスチャ特徴/HOG
/SIFT/HLAC/CHLAC/等

クラスタリング > k-平均法 (k-means clustering)

入力：特徴空間の点群 p_i , クラス数 k

- 各点 p_i にクラスIDをランダムに割り当てる
- クラスタ中心 c_j をクラスタの重心に移動
- 各点 p_i を中心 c_j が最も近いクラスに割り当てる
- 変化がなくなるまで2,3を繰り返す

クラス数 $k = 2$



クラスタリング > k-平均法 (k-means clustering)

◆ 利点

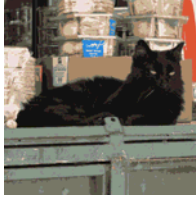
アルゴリズムが単純で実装が楽
サイズの小さなクラスタ分割が行える

◆ 欠点

初期割り当てに結果が依存
多様なクラスタ形状を扱えない
クラスタ数 k が既知の必要あり



入力画像



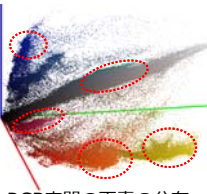
RGB空間



k=10

Bilateral空間

クラスタリング > まとめ



RGB空間の画素の分布



12領域

画素を**特徴空間に射影**し、その特徴空間内で**密度の濃い部分**を同一領域として分割する

- 特徴空間の選択とクラスタの発見法が大切
- 教師無し（正解データセット無し）学習の一種
- k-平均法, Mean Shift法 を紹介した

クラスタリング > Mean-Shift法

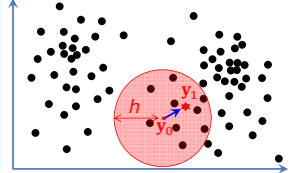
Mean Shift Procedure (MSP)

点 y_0 付近の点群密度の局所最大点を発見

入力：点群 p_i , 初期点 y_0 , バンド幅 h

- $y_{k+1} \leftarrow \frac{\sum_{i=1}^N g_i p_i}{\sum_{i=1}^N g_i}$ $g_i = \begin{cases} 1 & ||p_i - y_k|| \leq h \\ 0 & ||p_i - y_k|| > h \end{cases}$
- $||y_{k+1} - y_k|| < \text{閾値}$ まで 1 を繰り返す

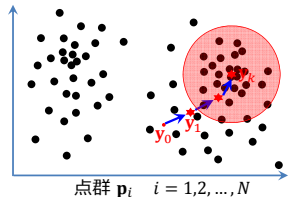
特徴空間の点群 p_i $i = 1, 2, \dots, N$



Mean Shift Clustering

方法1. 各画素位置 p_i からMSPを行い、近い点に収束した画素を同一クラスタにする

方法2. 特徴空間内に格子状に配置した点群 x_i にMSP行う。同じ点に収束するカーネルが通った画素を同一クラスタにする



Contents -画像領域分割-

デジタル画像データ

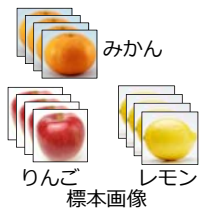
- Vector Graphics と Raster Graphics
- 標本化と量子化
- 階調数とヒストグラム
- ダイナミックレンジ とHDR

画像領域分割法

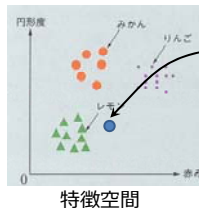
- 画像領域分割とは
- 閾値法
- 領域成長法
- クラスタリング
- クラス分類器
- 動的輪郭モデル
- グラフカット法
- 陰関数曲面再構成法

クラス分類器
Classifier

クラス分類器 > 概要



標準画像



特徴空間

入力画像

© CG-Arts協会,
デジタル画像処理, 2004

ラベル付けされた標準画像から分類法則を学習し、新たな画像にラベルを付ける手法

- 標準画像を特徴空間に射影する
- 入力画像を特徴空間に射影し、標準画像との距離や特徴空間の分割(サポートベクタマシン)を利用して、入力画像にラベルをつける

→ お勧めの書籍：『わかりやすいパターン認識』『PRML』

Contents -画像領域分割-

デジタル画像データ

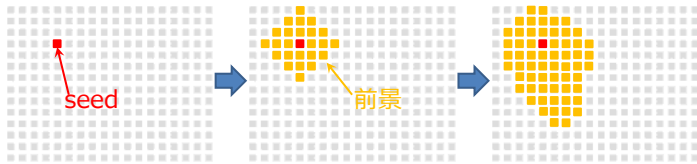
- Vector Graphics と Raster Graphics
- 標本化と量子化
- 階調数とヒストグラム
- ダイナミックレンジ とHDR

画像領域分割法

- 画像領域分割とは
- 閾値法
- クラスタリング
- クラス分類器
- 領域成長法
- 動的輪郭モデル
- グラフカット法
- 陰関数曲面再構成法

領域成長法
Region Growing

領域成長法 > 概要



- Seed画素から領域を徐々に成長させる (Seedは手で与えるか自動生成する)
- 局所的な規則に従って成長を止める
- Seed配置・成長規則について多くの研究がされている

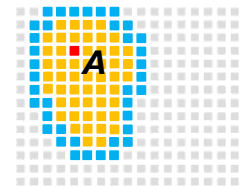
Adams R. et. al.: Seeded region growing. *IEEE PAMI* 16, 641-647, 1994.
 Roerdink J.B.T.M., et. al.: The Watershed Transform: Definitions, Algorithms and Parallelization Strategies, 2000.

領域成長法 > 二値化

TRegionGrowing.exe

■ Seed ■ 境界画素 T

■ 現在の領域 A



kステップ後の状態

領域成長法(二値化)

入力：複数のseed画素

- Seed画素を前景領域に追加
- 前景領域に隣接する画素 x のうち次式を満たすもの前景領域に追加
 $|c(seed) - c(x)| < r$ ※
- 成長が止まるまで(2)を繰り返す

$c(seed)$: seedの画素値

$c(x)$: 画素 x の画素値

r : パラメータ

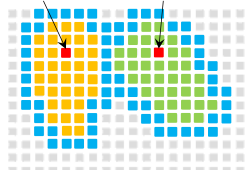
※追加条件は様々なものが考えられる

領域成長法 > 多値化

[Adams et. al. 1994]

■ Seed ■ 領域境界画素 T
 ■ 領域 A1 ■ 領域 A2

seed A1 seed A2



図はkステップ後の状態

Seeded Region Growing

入力：領域ID($A1, \dots, An$)の付いたSeed

- 各Seedを領域 $A1, \dots, An$ の要素とする
- 境界画素 x とその隣接領域 Ai のうち、次式が最小となる x を Ai に追加
 $\delta(x) = |c(x) - c(Ai)|$
- 全画素を追加するまで(2)を繰り返す

$c(x)$: x の色

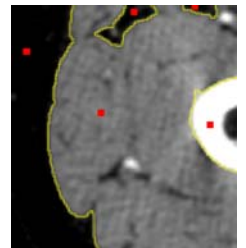
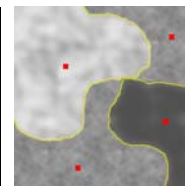
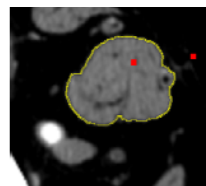
$c(Ai)$: x が隣接する領域 Ai の平均色

※2.において x が複数領域に隣接する場合は境界ラベル(-1など)をつける

領域成長法 > Seeded Region Growingの特徴

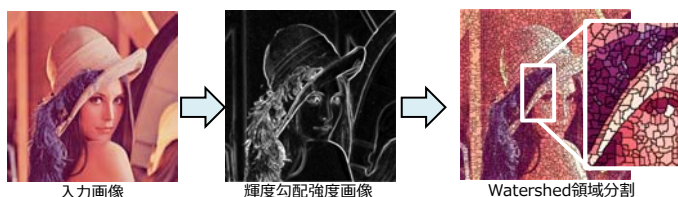
TRegionGrowing.exe

一様な画素値を持つ領域の分割に適する
 ぼけた境界では成長が止まりにくい



領域成長法 > Watershed Algorithm

[Roerdink J.B.T.M., et. al.: 2000.]

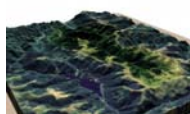


入力画像

輝度勾配強度画像

Watershed領域分割

勾配強度を高さと見なすと、勾配強度画像を地形と見なせる
 この地形の分水界を境界とする領域分割法

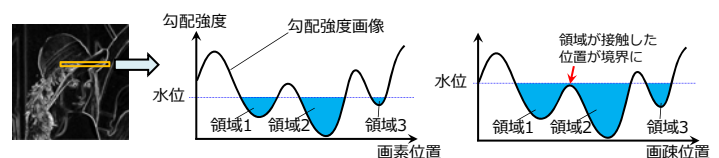


Watershed : 地形学における『分水界』を表す用語
 ある地形のある点に落ちた雨がどこに溜まるかを考える
 隣接しながらも溜まる先が異なる2点間を領域の境界に
 左図はwikipediaより

領域成長法 > Watershed Algorithm

TRegionGrowing.exe

[Roerdink J.B.T.M., et. al.: 2000.]

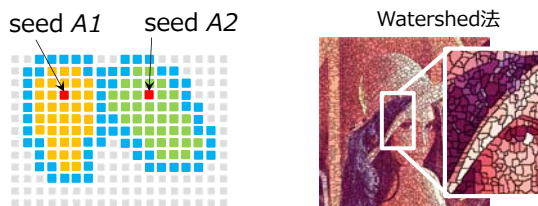


勾配強度を高さと考え水を下から満たしていく
 徐々に水位を上げ隣接領域が接した部分を境界とする

領域成長法の言葉で言うと...

- 勾配強度の全ての局所最小点にSeedを配置
- 勾配強度の高い部分で境界が止まるように、全領域を同時に成長させる

領域成長法 > まとめ



局所的な規則に従って領域を成長させる手法

Seed配置・成長規則に関する研究がなされている

『二値化法』『Seeded Region Growing法(多値化)』『Watershed法』を紹介

Adams R. et. al.: Seeded region growing. *IEEE PAMI* 16, 641-647, 1994.
Roerdink J.B.T.M., et. al.: The Watershed Transform: Definitions, Algorithms and Parallelization Strategies, 2000.

Contents -画像領域分割-

デジタル画像データ

- Vector Graphics と Raster Graphics
- 標本化と量子化
- 階調数とヒストグラム
- ダイナミックレンジとHDR

画像領域分割法

- 画像領域分割とは
- 閾値法
- 領域成長法
- クラスタリング
- 識別器
- 動的輪郭モデル
- グラフカット法
- 陰関数曲面再構成法

動的輪郭モデル Active Contours

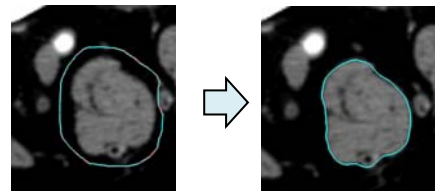
動的輪郭法 > 概要

領域境界を徐々に変形する

- 境界形状を滑らかに
- 境界が画像のエッジ（勾配強度の大きい部分）を通るよう

2手法に分類できる

- 境界を陽的に表現する手法: **Snakes法**
- 境界を陰的に表現する手法: **Level Set法**



CT画像は、理研・画像情報処理研究チームより

右図は
math.berkeley.edu/~sethian/
hianより

動的輪郭法 > Snakes (陽的表現) TActiveContour.exe

前提1. 曲線はパラメータ表現される $\mathbf{v}(s) = \begin{pmatrix} x(s) \\ y(s) \end{pmatrix}$ $s \in [0,1]$

前提2. 曲線のエネルギー

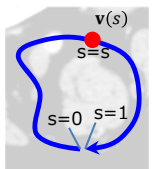
$$E(\mathbf{v}(s)) = \alpha E_{len} + \beta E_{curv} + \gamma E_{img}$$

$$E_{len} = \int_0^1 \left\| \frac{d\mathbf{v}(s)}{ds} \right\|^2 ds \quad : \text{弧長に対応する項 (長い} \rightarrow \text{大)}$$

$$E_{curv} = \int_0^1 \left\| \frac{d^2\mathbf{v}(s)}{ds^2} \right\|^2 ds \quad : \text{曲率に対応する項 (ガタガタ} \rightarrow \text{大)}$$

$$E_{img} = - \int_0^1 \| \nabla (G \otimes I(\mathbf{v}(s))) \| ds \quad : \text{画像(勾配強度)に対応する項}$$

↑
画像
ガウスフィルタ(ぼかす)
勾配強度を計算



課題. E を最小化する曲線 $\mathbf{v}(s)$ を探す (動的計画法 / 貪欲法など)

動的輪郭法 > Snakes > 折れ線近似

$$\mathbf{v}(s) = \begin{pmatrix} x(s) \\ y(s) \end{pmatrix} \quad s \in [0,1]$$

$$E_{len} = \int_0^1 \left\| \frac{d\mathbf{v}(s)}{ds} \right\|^2 ds$$

$$E_{curv} = \int_0^1 \left\| \frac{d^2\mathbf{v}(s)}{ds^2} \right\|^2 ds$$

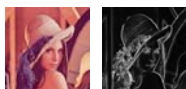
$$E_{img} = - \int_0^1 \| \nabla (G \otimes I(\mathbf{v}(s))) \| ds$$

折れ線近似

$$E_{len} \approx \sum_{i=1}^N (\mathbf{v}_i - \mathbf{v}_{i-1})^2$$

$$E_{curv} \approx \sum_{i=1}^N (\mathbf{v}_{i+1} + \mathbf{v}_{i-1} - 2\mathbf{v}_i)^2$$

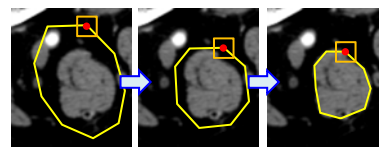
$$E_{img} \approx - \sum_{i=1}^N \| I'(\mathbf{v}_i) \|$$



※ $I'(\cdot)$ は元画像を平滑化した後、勾配強度を計算した画像

※上式は連結部分 \mathbf{v}_1 と \mathbf{v}_N を考慮していない

動的輪郭法 > Snakes > 貪欲法



逐次的に全頂点を移動し曲線を変形する

頂点 \mathbf{v}_i が寄与するエネルギー

$$E = \frac{\alpha}{2} (\mathbf{v}_i - \mathbf{v}_{i-1})^2 + \frac{\alpha}{2} (\mathbf{v}_{i+1} - \mathbf{v}_i)^2 \quad (E_{len})$$

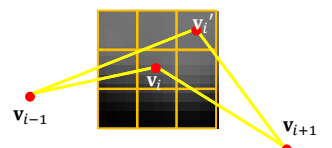
$$+ \beta (\mathbf{v}_{i+1} + \mathbf{v}_{i-1} - 2\mathbf{v}_i)^2 \quad (E_{curv})$$

$$- \gamma I'(\mathbf{v}_i) \quad (E_{img})$$

Snakes (貪欲法)

入力: 画像と初期曲線 (折れ線)

1. 各頂点 \mathbf{v}_i を局所エネルギーが最小となる近傍画素の中心に移動
2. 全頂点の総移動距離が閾値以下になるまで(1)を繰り返す



1-1 \mathbf{v}_i を近傍画素中心に移動してみる
1-2 局所エネルギー最小の画素に移動

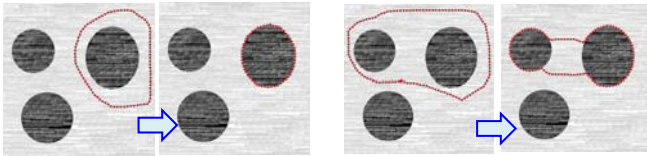
動的輪郭法 > Snakes

利点

- ・ ノイズに強い領域分割が可能
- ・ 高速かつ実装が簡単

欠点

- ・ パラメータに強く依存
- ・ 初期輪郭線に強く依存
- ・ トポロジー変化が困難



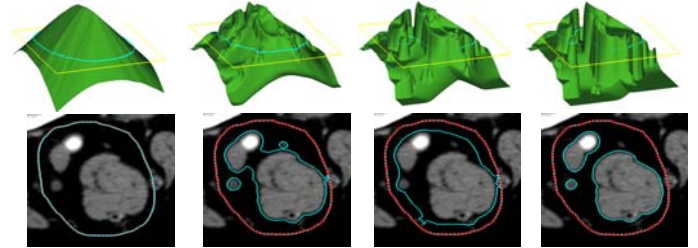
動的輪郭法 > Level set法 (陰的表現)

TActiveContour.exe

SKIP??

曲線を符号付きスカラー場（内挿関数）のゼロ等高線で表現
スカラー場の初期値は初期境界からの距離場を利用
スカラー場を変化させることで曲線を間接的に変形する

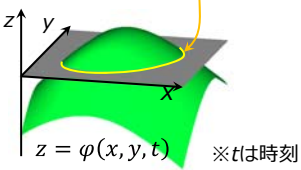
→ トポロジー変化に対応できる



動的輪郭法 > Level Set法 (陰的表現)

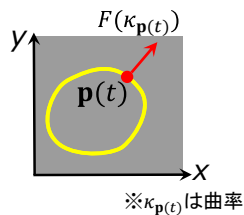
前提1. 境界曲線を内挿関数 $\varphi(x, y, t)$ のゼロセットで表現

曲線: $\varphi(p_x(t), p_y(t), t) = 0$



※tは時刻

前提2. 曲線上の点 $p(t)$ は法線方向に速度 $F(\kappa_{p(t)})$ で動く



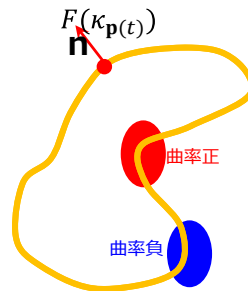
※ $\kappa_{p(t)}$ は曲率

課題. 収束するまで曲線を変形する

※ 実際に変化させるのは $\varphi(x, y, t)$

※ 移動速度 $F(\kappa_{p(t)})$ をうまく指定すると、輪郭の進展が境界で止まる

動的輪郭法 > Level Set法



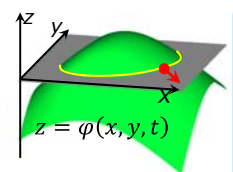
速度 F を曲率 $\kappa_{p(t)}$ の関数にする理由

曲線の凸部・凹部で挙動を変化できる
例) 凹部は外へ、凸部は内へ向かって移動

曲がり具合に応じて挙動を変化できる
例) 曲がり具合が大きいところは速く移動

※曲率の正負は曲線の向き・定義に依存

動的輪郭法 > Level Set法



問題の前提

点 p が曲線に乗る $\varphi(p_x(t), p_y(t), t) = 0$

点 p は法線 n 方向に速度 F で移動 $\frac{\partial p(t)}{\partial t} = F(\kappa_{p(t)})n$

法線 n は φ の勾配 $n = \frac{\nabla \varphi}{\|\nabla \varphi\|}$

内挿関数 φ の時間進展に関する方程式を導く

整理する $\frac{\partial \varphi(p_x(t), p_y(t), t)}{\partial t} = -F(\kappa_{p(t)})\|\nabla \varphi\|$ ※この式は曲線上で成立
※変形の詳細は付録へ

全体に拡張 $\frac{\partial \varphi(x, y, t)}{\partial t} = -F(\kappa)\|\nabla \varphi\|$ ※ κ は内挿関数 φ の曲率

位置 (x, y) と時刻 t について離散化 $\frac{\varphi(i, j, t+1) - \varphi(i, j, t)}{h} = -F(\kappa_{i,j})\|\nabla \varphi(i, j, t)\|$ ※ i, j は画素位置
※ h は微小時刻

動的輪郭法 > Level Set法

TActiveContour.exe

1. 初期曲線から初期スカラー場 $\varphi(i, j, 0)$ を構築
(i, j は画素位置, $\varphi(i, j, 0)$ は初期曲線からの距離場)

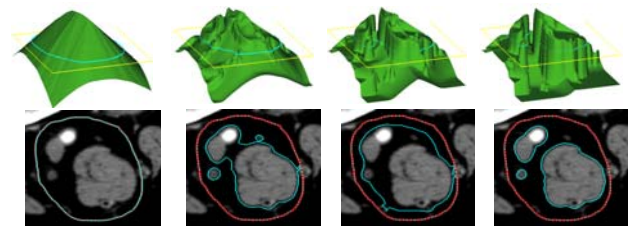
2. 変化が十分小さくなるまで時刻 t を進め φ を更新

$$\frac{\varphi(i, j, t+1) - \varphi(i, j, t)}{h} = -\frac{a - b\kappa_{i,j}}{1 + \|\nabla(G \otimes I(i, j))\|} \|\nabla \varphi\|$$

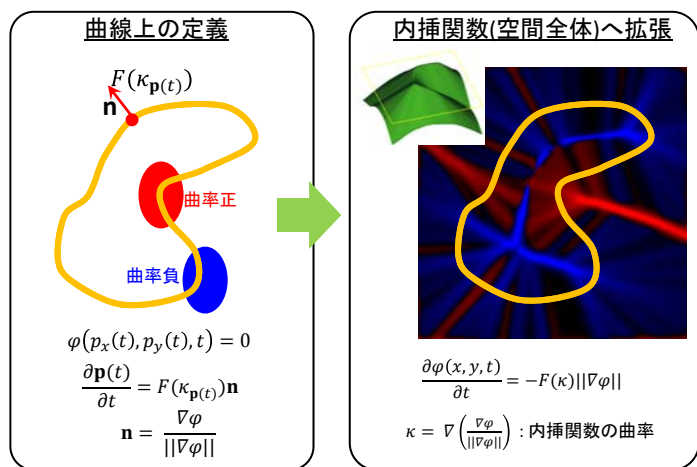
a, b はパラメータ

$\kappa_{i,j} > a/b$ なら正方向に進む
エッジ部分では変化が遅い

3. $\varphi(i, j, t) = 0$ である画素を境界として出力する



動的輪郭法 > Level set法



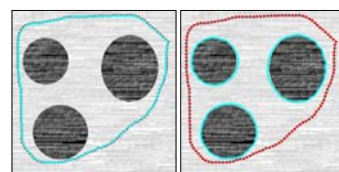
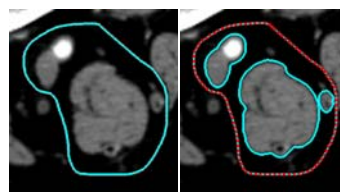
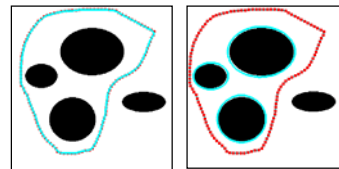
動的輪郭法 > Level set法

利点

- ノイズに対し堅固・高速
- トポロジー変化に対応：曲線の分離・結合を自然に表現できる

欠点

- 速度 $F(\kappa_{\mathbf{p}(t)})$ の適切な選択が難しい
- パラメータ調整が大変

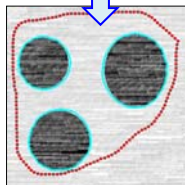
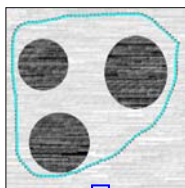


動的輪郭法 > まとめ

境界曲線を『形状が滑らかになるように』『画像のエッジを通る様に』変形する手法

- 境界を陽的に表現する：Snakes 法
- 境界を陰的に表現する：Level Set 法

変形モデル(速度の定義)・最適解の計算法・高次元化など、関連研究は多い



Contents -画像領域分割-

デジタル画像データ

- Vector Graphics と Raster Graphics
- 標準化と量子化
- 階調数とヒストグラム
- ダイナミックレンジ と HDR

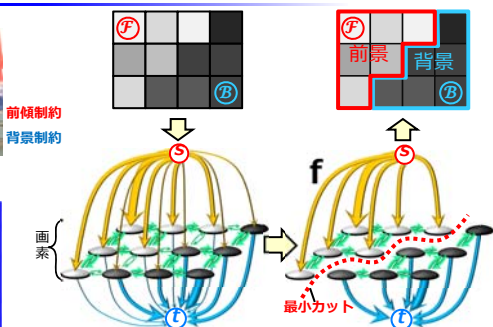
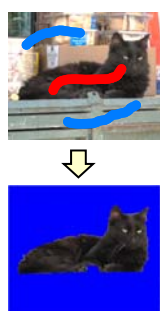
画像領域分割法

- 画像領域分割とは
- 閾値法
- 領域成長法
- クラスタリング
- 識別器
- 動的輪郭モデル
- **グラフカット法**
- 陰関数曲面再構成法

グラフカット法 Graph Cut Image Segmentation

グラフカット法 > 概要

TGraphCut.exe



前景・背景制約画素をヒントにして画像を二値化する
二値化をエネルギー最小化問題として定式化し、フローネットワークの最小カットにより最適解を計算する

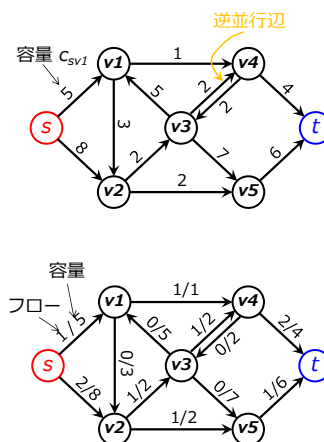
グラフカット法 > フローネットワーク

フローネットワーク

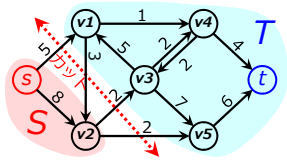
- 容量付き有向グラフ $G = (V, E)$
- 頂点集合 V と辺集合 E から成る
- 辺 (p, q) は容量 $c_{pq} > 0$ を持つ
- 始点 s と終点 t を含む

フロー

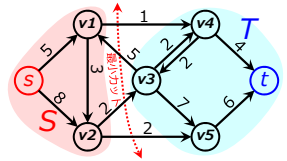
- 各辺 (p, q) には容量 c_{pq} を超えない範囲でフロー f_{pq} が流れる： $0 \leq f_{pq} \leq c_{pq}$
- 『頂点 v に流入するフロー』と『頂点 v から流出するフロー』は等しい
- **総流量**：始点から出るフローの総和
- ネットワークに流せる総流量が最大のフローを**最大フロー**と呼ぶ



グラフカット法> フローネットワーク



カット : $S=\{s, v2\}$, $T=\{v1, v3, v4, v5, t\}$
 カットセット : $\{(s, v1), (v1, v2), (v2, v3), (v2, v5)\}$
 カット容量 : $5 + 2 + 2 = 9$



最小カット : $S=\{s, v1, v2\}$, $T=\{v3, v4, v5, t\}$
 カット容量 : $1 + 2 + 2 = 5$

カット:

頂点を『sを含む部分集合S』と『tを含む部分集合T』に分割する

カットセット:

部分集合SとTの間をつなぐ辺の集合

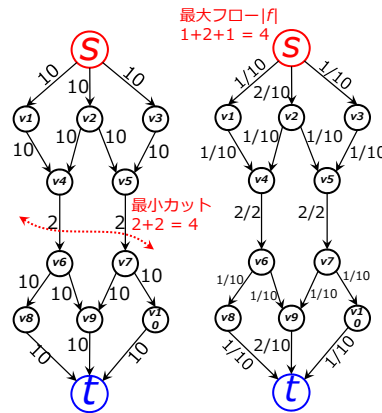
カットの容量:

カットセットのうち $S \rightarrow T$ 方向の辺の容量総和 (逆向きの辺は無視する)

最小カット

容量が最小となるカット

グラフカット法> フローネットワーク



最大フロー-最小カットの定理

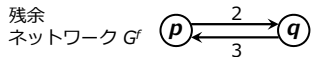
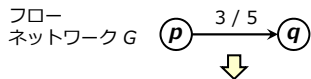
任意のフローネットワークについて、その最大フローと最小カットは等しい
 最大フローはネットワークの一番細い部分(最小カット)によって決定される

※ 最大フローが流れているとき、始点sから不飽和辺のみを使って到達できる頂点群をSとし、 $T = V - S$ とすると、S-Tは最小カットをなす

※ 最大フロー・最小カットの探索には様々なアルゴリズムが存在し、“解ける”

グラフカット法> フローネットワーク

SKIP ?

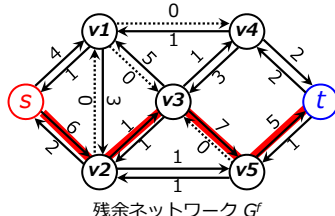
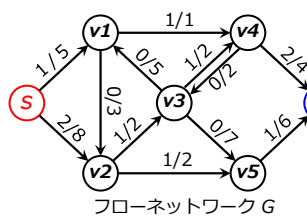


残余ネットワーク G'

G にフローが流れているとき、『フローの可変範囲』を表すネットワーク

増加可能経路

残余ネットワーク内の $s \rightarrow t$ 経路



グラフカット法> フローネットワーク

SKIP

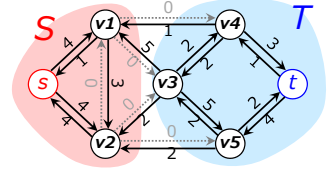
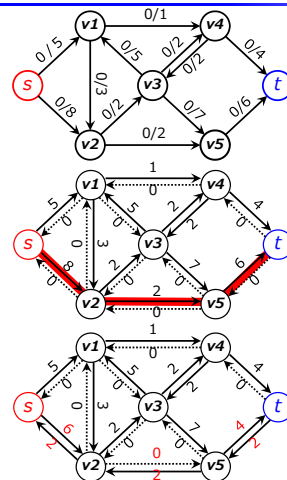
最大フローアルゴリズム

入力: フローネットワークG

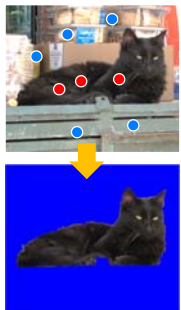
出力: 最大フローと最小カット

1. フローを0で初期化
2. 残余ネットワークを構築
3. 経路Pが無くなるまで以下を繰り返す
 - 3-1) 増加可能経路 P の探索
 - 3-2) 経路Pに沿ってフロー追加

※ 増加可能経路 P : 残余ネットワーク内の $s - t$ 間をつなぐ単純経路



グラフカット法 > 問題の定義



入力: 画像、制約画素集合 (前景 \mathcal{F} ・背景 \mathcal{B})
 出力: 以下を満たす二値化

- ・制約画素 p は必ず制約を満たす
- ・非制約画素 p は、その特徴(色など)が前景画素 \mathcal{F} に近ければ前景に、 \mathcal{B} に近ければ背景になる
- ・境界は特徴の異なる画素間を通る

エネルギー最小化問題として定式化

$$\operatorname{argmin}_{\{L_p | p \in \Omega\}} \sum_{p \in \Omega} E_1(L_p) + \lambda \times \sum_{(p, q) \in \mathcal{N}} E_2(L_p, L_q)$$

Ω : 全画素集合

\mathcal{N} : 近傍画素集合

L_p : 画素 $p \in \Omega$ につくラベル値 $\{\text{fore}, \text{back}\}$ のどちらか

グラフカット法> エネルギー最小化

$$\operatorname{argmin}_{\{L_p | p \in \Omega\}} \sum_{p \in \Omega} E_1(L_p) + \lambda \times \sum_{(p, q) \in \mathcal{N}} E_2(L_p, L_q)$$

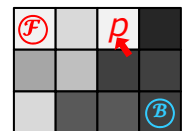
Ω : 全画素集合
 \mathcal{N} : 近傍画素集合
 L_p : 画素 $p \in \Omega$ につくラベル値 $\{\text{fore}, \text{back}\}$ のどちらか

$E_1(L_p)$: 『データ項』画素 p のラベル付の不正確さに反応する項

画素 p が前景制約画素に似ているなら...

$$E_1(L_p = \text{fore}) \leftarrow \text{小}$$

$$E_1(L_p = \text{back}) \leftarrow \text{大}$$



$$E_1(L_p = \text{fore}) = \begin{cases} 0 & p \in \mathcal{F} \\ \infty & p \in \mathcal{B} \\ t_p^{\text{fore}} & \text{other} \end{cases}$$

$$E_1(L_p = \text{back}) = \begin{cases} 0 & p \in \mathcal{F} \\ \infty & p \in \mathcal{B} \\ t_p^{\text{back}} & \text{other} \end{cases}$$

$$t_p^{\text{fore}} = \frac{d_p^{\mathcal{F}}}{d_p^{\mathcal{F}} + d_p^{\mathcal{B}}}, t_p^{\text{back}} = \frac{d_p^{\mathcal{B}}}{d_p^{\mathcal{F}} + d_p^{\mathcal{B}}}$$

$$d_p^{\mathcal{F}} = \min_{k \in \mathcal{F}} \|c_p - c_k\|$$

$$d_p^{\mathcal{B}} = \min_{k \in \mathcal{B}} \|c_p - c_k\|$$

※ c_p は画素 p の画素値

※ t_p^{fore} は画素 p の色が前景画素に似るほど小さくなるよう指定する

グラフカット法> エネルギー最小化

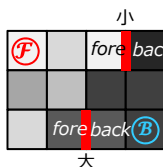
$$\operatorname{argmin}_{\{L_p \mid p \in \Omega\}} \sum_{p \in \Omega} E_1(L_p) + \lambda \times \sum_{(p,q) \in \mathcal{N}} E_2(L_p, L_q)$$

Ω : 全画素集合
 \mathcal{N} : 近傍画素集合
 L_p : 画素 $p \in \Omega$ につくラベル値 $\{fore, back\}$ のどちらか

$E_2(L_p, L_q)$: 『平滑化項』 隣り合う画素が似た特徴 (色) を持つときは、なるべく同じラベルをつける

☆隣接画素 p, q に同じラベルをつける
 $\rightarrow E_2 = 0$

☆隣接画素 p, q に違うラベルをつける
 p, q が似た色を持つ $\rightarrow E_2$ は大
 p, q が遠い色を持つ $\rightarrow E_2$ は小



$$E_2(L_p, L_q) = \begin{cases} 0 & L_p = L_q \\ \frac{1}{1 + \|\mathbf{c}_p - \mathbf{c}_q\|} & \text{other} \end{cases}$$

\mathbf{c}_p は画素 p の画素値

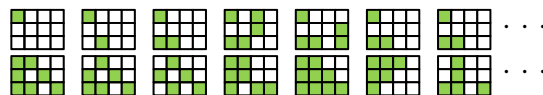
グラフカット法> エネルギー最小化

$$\operatorname{argmin}_{\{L_p \mid p \in \Omega\}} \sum_{p \in \Omega} E_1(L_p) + \lambda \times \sum_{(p,q) \in \mathcal{N}} E_2(L_p, L_q)$$



この問題は解くのが難しい

- 可能なラベリングの組み合わせが多い (3×4画像でも $2^{12} = 4096$ 通り)
- 局所最小解が多い
- グラフカット以前はランダムウォークを利用 (『焼きなまし法』)



『この問題の大域最小化解は、フローネットワークの最小カットにより高速に求められる』 [Boykov Y., Jolly M-P. MICCAI, 276-286, 2000.]

※ L_p が二状態をとる場合 (二値化) に限る

グラフカット法> グラフを用いたエネルギー最小化

$$\operatorname{argmin}_{\{L_p \mid p \in \Omega\}} \sum_{p \in \Omega} E_1(L_p) + \lambda \times \sum_{(p,q) \in \mathcal{N}} E_2(L_p, L_q)$$

画像からフローネットワークを構築

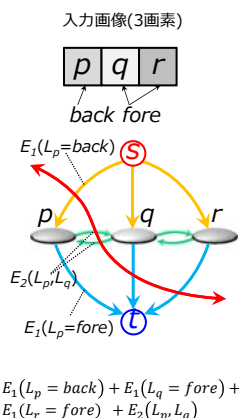
- 頂点 V : 全画素, 始点 s , 終点 t
- 辺 E : 右図

フローネットワークをカットし

- s に連結する画素にラベル *fore* をつける
- t に連結する画素にラベル *back* をつける

カット容量 = エネルギー

エネルギー最小化 = 最小カット



$$E_1(L_p = back) + E_1(L_q = fore) + E_1(L_r = fore) + E_2(L_p, L_q)$$

グラフカット法

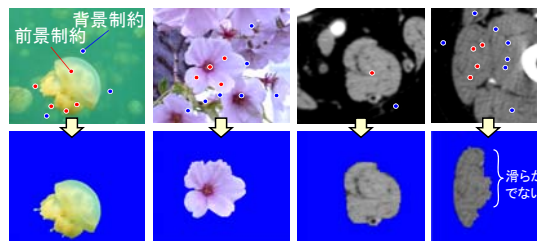
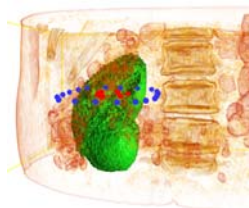
VoTracer.exe

利点

- 高速・高精度
- 高次元化が容易
- UIと相性がよい

欠点

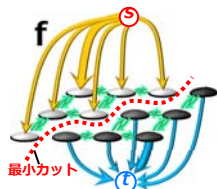
- 境界が不明瞭な領域には利用し難い
- 血管・筋膜等、細い・薄い形状には不向き



グラフカット法 > まとめ

$$\operatorname{argmin}_{\{L_p \mid p \in \Omega\}} \sum_{p \in \Omega} E_1(L_p) + \lambda \times \sum_{(p,q) \in \mathcal{N}} E_2(L_p, L_q)$$

Ω : 全画素集合
 \mathcal{N} : 隣接画素集合
 L_p : 画素 p に付けるラベル値 $\{fore, back\}$



画像二値化をエネルギー最小化問題として定式化し、フローネットワークの最小カットにより最適解を計算する
 境界が明瞭な領域の分割にはかなり良い結果を出力できる
 生体画像の領域分割に向く

Contents -画像領域分割-

デジタル画像データ

- Vector Graphics と Raster Graphics
- 標本化と量子化
- 階調数とヒストグラム
- ダイナミックレンジ と HDR

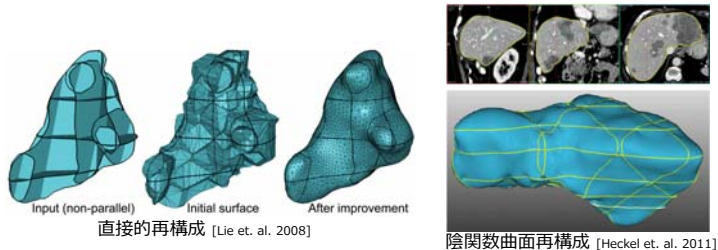
画像領域分割法

- 画像領域分割とは
- 閾値法
- 領域成長法
- クラスタリング
- 識別器
- 動的輪郭モデル
- グラフカット法
- 陰関数曲面再構成法

曲面再構成法
 Surface Reconstruction

曲面再構成法 > 概要

- 形状モデリングのための技術を応用
- 輪郭線制約から三次元境界曲面を生成する
- 直接的（陽的）曲面再構成法 と 陰関数曲面再構成法

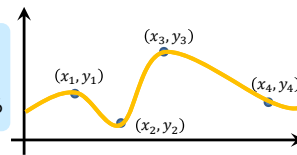


TURK G., O'BRIEN J. F.: Modelling with implicit surfaces that interpolate. ACM TOG 21, 4(2002), 855-873.
 Heckel F., et. al.: Interactive 3D medical image segmentation with energy-minimizing implicit function. VCBM 35, 2(2011), 275-287.
 Liu L. et. al.: Surface reconstruction from non-parallel curve networks. CGF 27, 2(2008), 155-163.
 Takashi Ijiri, et. al. Bilateral Hermite Radial Basis Functions for Contour-based Volume Segmentation. CGF, 2013.

曲面再構成法 > 放射基底関数補間法 1次元

補間法

入力: N 点 x_i における値 $y_i, i=1, 2 \dots N$
 問題: $f(x_i) = y_i$ を満たす関数 f を求める



放射基底関数補間法

$$f(x) = \sum_{i=1}^N \alpha_i \varphi(|x - x_i|) + ax + b$$

$$\sum_{i=1}^N \alpha_i = 0, \quad \sum_{i=1}^N \alpha_i x_i = 0$$

$$\varphi(t) = t^3 : \text{カーネル}$$

$f(x_i) = y_i$ を代入し未知数 α_i, a, b を求める.

$N=4$ ならば... 未知数は4+2個
 $f(x) = \alpha_1 |x - x_1|^3 + \alpha_2 |x - x_2|^3 + \alpha_3 |x - x_3|^3 + \alpha_4 |x - x_4|^3 + ax + b$

制約は4+2個
 $f(x_1) = y_1, f(x_2) = y_2, f(x_3) = y_3, f(x_4) = y_4,$
 $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 0$
 $\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_4 x_4 = 0$
 連立一次方程式

曲面再構成法 > 放射基底関数補間法 d 次元

補間法

入力: N 点 $\mathbf{x}_i \in R^d$ における値 y_i
 問題: $f(\mathbf{x}_i) = y_i$ を満たす関数 f を求める

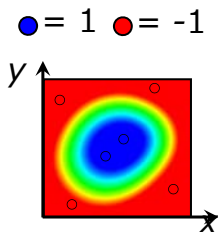
放射基底関数補間法

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i \varphi(|\mathbf{x} - \mathbf{x}_i|) + \mathbf{a} \cdot \mathbf{x} + b$$

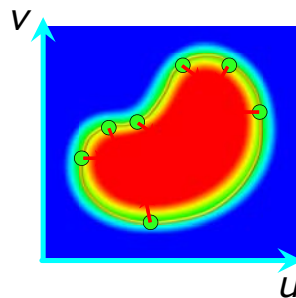
$$\sum_{i=1}^N \alpha_i = 0, \quad \sum_{i=1}^N \alpha_i \mathbf{x}_i = 0$$

$$\varphi(t) = t^3 : \text{tri-harmonic カーネル}$$

$f(\mathbf{x}_i) = y_i$ を代入し未知数 α_i, \mathbf{a}, b を求める.



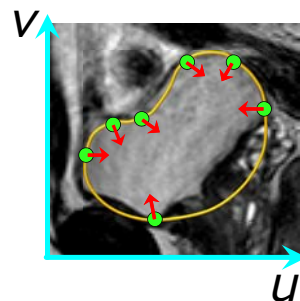
曲面再構成法 > 陰関数曲面再構成



- 境界の通る点・法線を指定
- uv 空間にスカラー場 f を構築
 $f(x) = 0, \nabla f(x) = n$
- スカラー場のゼロセットを抽出

図は[Ijiri et al. Bilateral Hermite Radial Basis Function... EUROGRAPHICS 2013]より

曲面再構成法 > 陰関数曲面再構成法の課題

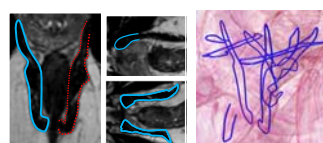


画像のエッジを追従しない

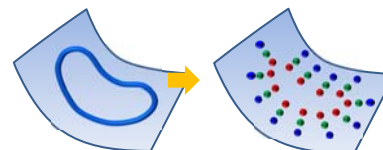
- 動的輪郭モデルで境界を動かす [Aliroteh M et. al. 2007]
- Bilateral空間への拡張

図は[Ijiri et al. Bilateral Hermite Radial Basis Function... EUROGRAPHICS 2013]より

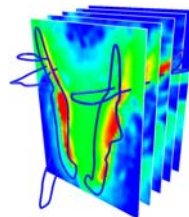
曲面再構成法 > 陰関数曲面再構成法



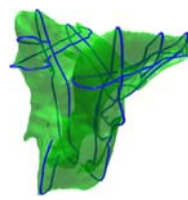
入力: 三次元空間内の輪郭線



1) 輪郭線をサンプリングし制約点を生成



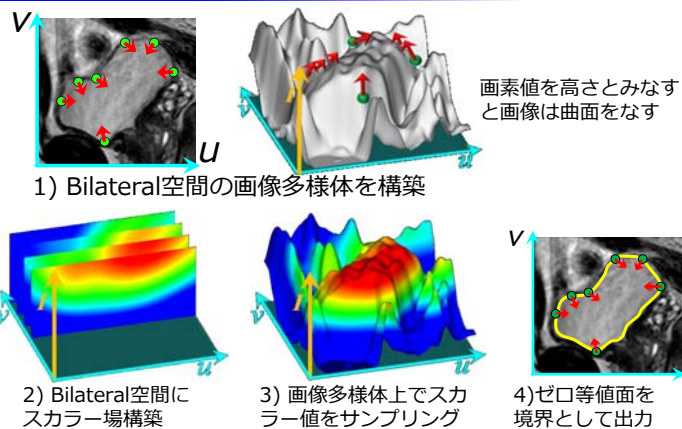
2) 放射基底関数補間により滑らかなスカラー関数 $f(\mathbf{x})$ 生成



3) 関数 $f(\mathbf{x})$ のゼロ等値面を境界として出力

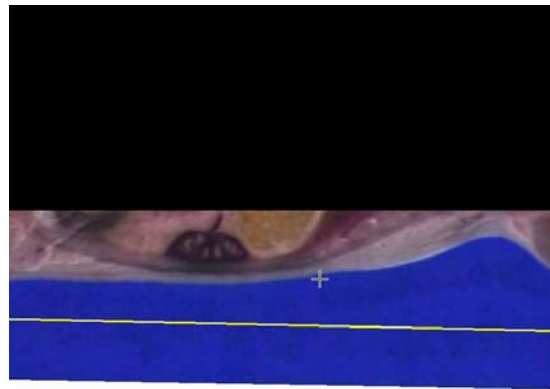
図は[Ijiri et al. Bilateral Hermite Radial Basis Function... EUROGRAPHICS 2013]より

曲面再構成法 > 陰関数曲面再構成法 (Bilateral空間に拡張)



図は[Ijiri et al. Bilateral Hermite Radial Basis
Function... EUROGRAPHICS 2013]より

曲面再構成法 > 陰関数曲面再構成法 (Bilateral空間に拡張)



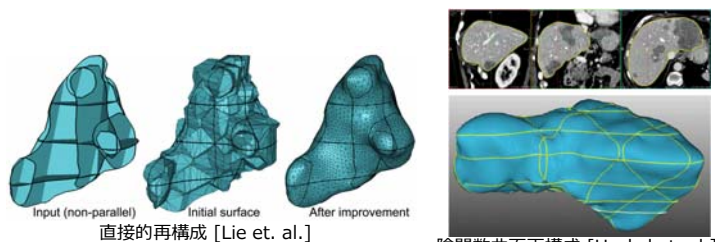
図は[Ijiri et al. Bilateral Hermite Radial Basis Function...
EUROGRAPHICS 2013]より

曲面再構成法 > まとめ

輪郭線制約から三次元境界曲面を生成する

直接的曲面再構成：輪郭線頂点を直接つなぎ境界面を構築

陰関数曲面再構成：輪郭線から滑らかなスカラー場を構築し、
そのゼロ等値面を出力



TURK G., O'BRIEN J. F.: Modelling with implicit surfaces that interpolate. ACM TOG 21, 4(2002), 855-873.
Heckel F., et. al.: Interactive 3D medical image segmentation with energy-minimizing implicit function. VCBM 35, 2(2011), 275-287.
Liu L. et. al.: Surface reconstruction from non-parallel curve networks. CGF 27, 2(2008), 155-163.
Takashi Ijiri, et. al. Bilateral Hermite Radial Basis Functions for Contour-based Volume Segmentation. CGF, 2013.

まとめ -画像領域分割-

デジタル画像データ

- Vector Graphics と Raster Graphics
- 標本化と量子化
- 階調数とヒストグラム
- ダイナミックレンジ とHDR

画像領域分割法

- 画像領域分割とは
- 閾値法
- 領域成長法
- クラスタリング
- 識別器
- 動的輪郭モデル
- グラフカット法
- 陰関数曲面再構成法

画像領域分割は重要な研究課題

確率統計、特徴抽出、微分方程式、
グラフ理論、補間法など多岐にわた
る技術が応用されている

まとめ -画像領域分割-

任意の画像（写真、CT、MRI、顕微鏡）、任意の関心領域(人物、臓器、腫瘍、細胞内小器官)に対し良い結果を出せる『オールマイティ』な領域分割法は未だ実現されていない

ユーザは、対象に応じて手法を注意深く選択することが大切
(選択肢が多くあることを知っておくだけでも)

後期は汎用ソフトで遊んでみる予定 (ImageJ / VoTracer)

バイオメディカルエンジニアリング

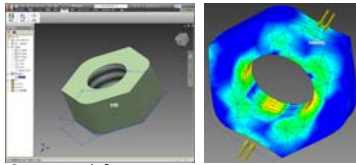
- 前期
 - 生体シミュレーションの概要と現状-医療応用
 - 生体シミュレーションの概要と現状-スポーツ応用
 - 画像取得技術
 - 生体材料の力学情報取得技術
 - 画像処理概論 - 画像領域分割の基礎
 - **形状モデリング概論 - 三次元モデルの表現法**
- 後期
 - 生体シミュレーション
 - 生体シミュレーション
 - 構造力学演習
 - 医療ロボットならびに人由来材料研究の倫理規定
 - **画像処理概論 - 画像処理の基礎 -**
 - **画像処理演習**

姫野先生
姫野先生
横田先生
横田先生
井尻
井尻

3Dモデルの応用



© Blender
エンターテイメント
映画・ゲーム・画像編集



ものづくり
設計(CAD)・シミュレーション(CAE)・製造(CAM)
機械・建築・土木・電気回路 等



手術シミュレータ (© 三菱プレジジョン)

計算科学 (医療・科学)
シミュレーションを通じ科学的発見を
血流・筋腱・細胞変形・手術シム 等

Contents

3D モデル表現

- 点群
- ワイヤーフレームモデル
- サーフェスモデル
- ソリッドモデル

SKIP?

3D モデリング

- 3D スキャナ
- CGソフト
- CADソフト
- 画像領域分割

まとめ

Contents

3D モデル表現

- 点群
- ワイヤーフレームモデル
- サーフェスモデル
- ソリッドモデル

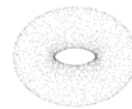
3D モデリング

- 3D スキャナ
- CGソフト
- CADソフト
- 画像領域分割

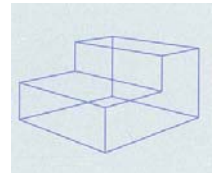
まとめ

3Dモデル表現

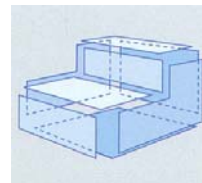
3Dモデル表現 > 概要



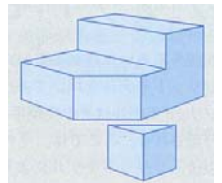
点群モデル(点)



ワイヤーフレームモデル(点・線)



サーフェスモデル(点・線・面)



ソリッドモデル(点・線・面・体積)

© CG-arts協会, ビジュアル情報処理

3Dモデル表現 > 点群モデル

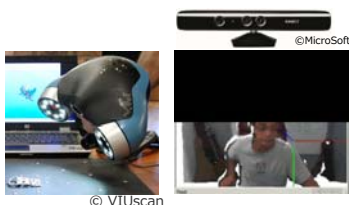
データ表現

メタ情報(位置・法線・色)を持つ点の集合

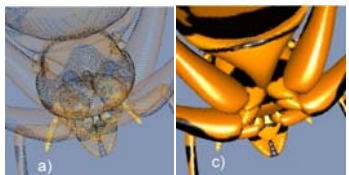
応用先と特徴

高速レンダリング

- ⊖ 内外判定不可
- ⊖ 利用性が悪い (CAD/CAM)



© VIUScan



Pfister P. et. al., Surfels: Surface Elements as Rendering Primitives, SIGGRAPH 2000

3Dモデル表現 > ワイヤーフレームモデル

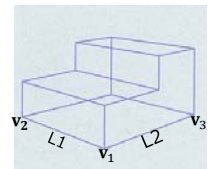
データ表現

ポリライン (頂点と連結関係を保持)

- 頂点: v_1, v_2, \dots, v_N
- 線分: $L1(1, 2), L2(1, 3), \dots$

パラメトリック曲線

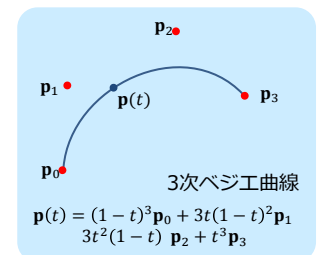
- 制約点群 p_i



応用先と特徴

CG初期の映画等でよく用いられた表示が高速

- ⊖ 隠れ面処理・内外判定不可
- ⊖ CAD/CAMへの利用性が悪い



3Dモデル表現 > サーフェスモデル

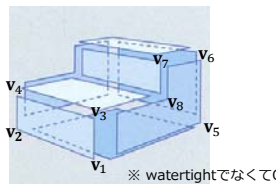
データ表現

ポリゴン (頂点群と連結関係を保持)

- 頂点: v_1, v_2, \dots, v_N
- 面: $S_1(1, 2, 4, 3), S_2(1, 3, 8, 7, 6, 5), \dots$

パラメトリック曲面

- 制約点群 p_{ij}



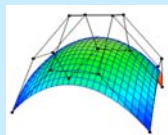
応用先と特徴

CGに頻りに利用される

シミュレーション(板・膜状物体)

- ⊖ 内・外情報なし
- ⊖ 体積表現不可

ベジエ曲面



[http://www.mizuno.org/gl/bs/index.ja.shtml]

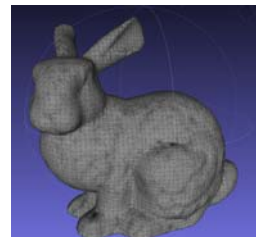
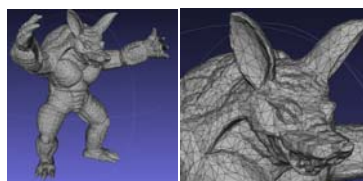
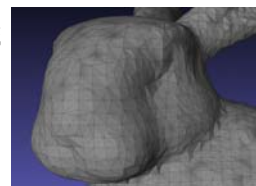
$$p(u, v) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} B_i^{N-1}(u) B_j^{M-1}(v) p_{ij}$$

B: ベルンシュタイン関数

3Dモデル表現 > サーフェスモデル

CGでは三角形ポリゴンモデルが主流

- 平面(法線)が一意に決まる
 - 高速
 - 任意曲面を位相によらず表現可能
- 現実の形状データを統合的に扱える



3Dモデル表現 > ソリッドモデル(点・線・面・体積)

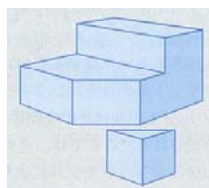
データ表現

境界表現

CSG表現

多面体メッシュ

八分木

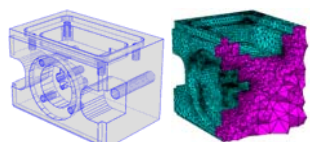


応用先と特徴

内外判定が可能 (体積・交差)

CAD・CAEで多様な応用

- ⊖ データ構造が複雑

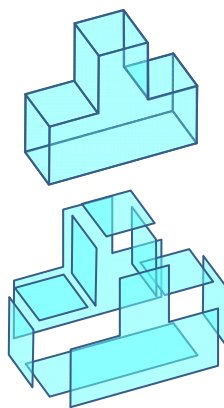


境界表現

四面体メッシュ

[wias-berlin.de/software/tetgen/]

3Dモデル表現 > ソリッドモデル > 境界表現



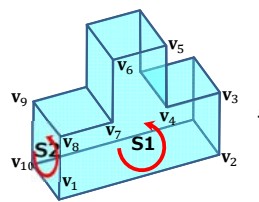
物体の境界を保持する

- 頂点・面 (N -角形)・辺
- **Watertight**なサーフェスモデル
- 常に内外判定可
- 体積計算可能
- 論理演算可能

面に裏表の概念有り

CADシステムでの主流

3Dモデル表現 > ソリッドモデル > 境界表現



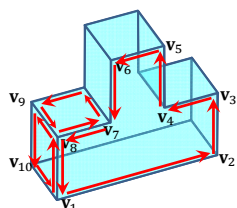
※外から見て反時計回りの頂点順序

データ構造 (プログラム内のデータの持ち方)

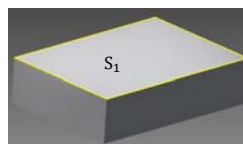
頂点V: 三次元ベクトルの配列を持つ
 $\{v_1, v_2, \dots, v_N\}$

面S: 各面を構成する頂点番号を持つ
 $\{S_1(1, 2, 3, 4, 5, 6, 7, 8), S_2(1, 8, 9, 10), \dots\}$

辺E: 各辺を2つの有向辺に分けた辺を持つ
 $\{e_1(1, 8), e_2(8, 1), e_3(1, 2), e_4(2, 1), \dots\}$
※ハーフエッジデータ構造



3Dモデル表現 > ソリッドモデル > 境界表現



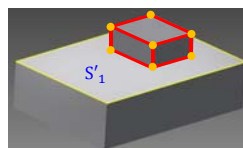
頂点V: $\{v_1, v_2, \dots, v_8\}$

面S: $\{S_1, S_2, \dots, S_6\}$

面E: $\{e_1, e_2, \dots, e_{12}\}$



変形操作が行われたら、Watertight性・辺や面の参照関係を保つように、頂点・面・辺情報を更新する
※プログラミングはそれなりに大変



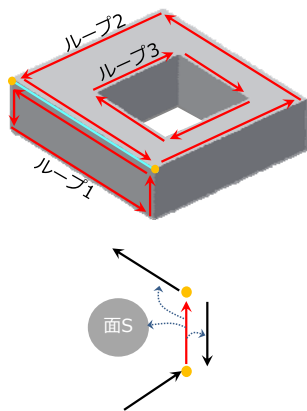
頂点V: $\{v_1, v_2, \dots, v_8, v_9, v_{10}, \dots, v_{16}\}$

面S: $\{S'_1, S_2, \dots, S_6, S_7, S_8, \dots, S_{11}\}$

面E: $\{e_1, e_2, \dots, e_{12}, e_{13}, e_{14}, \dots, e_{24}\}$

3Dモデル表現 > ソリッドモデル > 境界表現

SKIP



面ループ

面の境界の辺をつないだループ
穴の空いた面は面ループを複数持つ
各面にその面の面ループを関連づける

ハーフエッジ

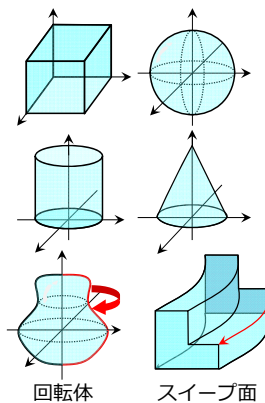
辺を二本の有向辺で表現するデータ構造
各辺は次の参照を持つ

- 2頂点
- 対をなす辺
- 自分の属す面
- 面ループ上の次辺

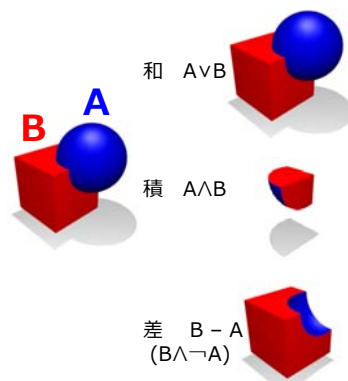
面ループ・ハーフエッジにより、頂点・面・辺の追加・削除など複雑な処理が可能に

3Dモデル表現 > ソリッドモデル > CSG表現

基本図形(プリミティブ)



論理(Boolean)演算



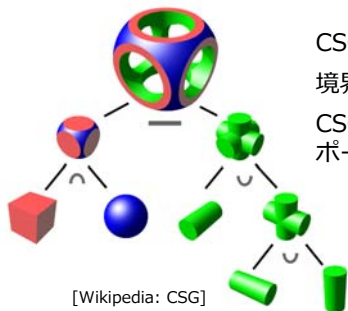
画像はWikipedia "CSG"より

3Dモデル表現 > ソリッドモデル > CSG表現

CSG : Constructive Solid Geometry

形状をプリミティブの論理演算の組み合わせで表現する

論理演算: 和・差・積・集合

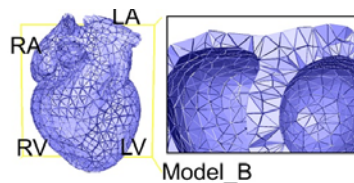


[Wikipedia: CSG]

CSG表現 → 境界表現 の変換は容易
境界表現 → CSG表現 の変換は困難
CSG表現は一部のCADソフトでサポートされる

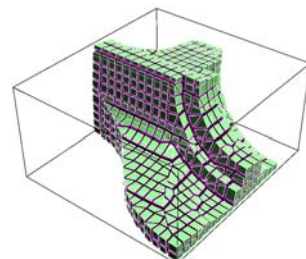
3Dモデル表現 > ソリッドモデル > 多面体メッシュ

- 物体を密に詰まった多面体で表現する
- シミュレーション に利用(構造解析・熱拡散)
- 有限要素メッシュとも呼ばれる
- ⊗ 歪んだ要素が避けにくい (特に表面付近)



四面体メッシュ

画像は理研 V-DualGrid より



六面体メッシュ

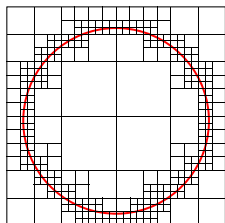
画像は理研 V-DualGrid より

3Dモデル表現 > ソリッドモデル > Octree(八分木)

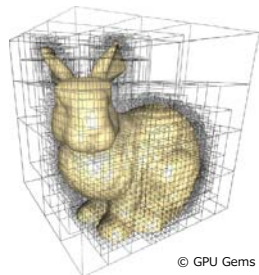
モデルの境界で細くなるよう空間全体を分割

(モデル表現というより空間表現)

- シミュレーション(差分法)に利用 (流体解析など)
- 交差判定に利用



Quad Tree(4分木)



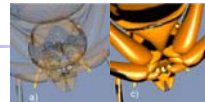
© GPU Gems 2

3Dモデル表現 > まとめ

点群モデル(点)

Point-based Graphics

Laser scannerデータとして



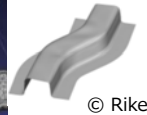
ワイヤーフレームモデル(点・線)

CG表現

サーフェスモデル(点・線・面)

三角形メッシュ表現 → CG

多角形メッシュ表現 → CAE (膜・板状要素)

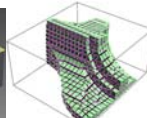
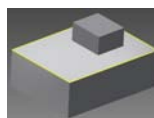


© Riker

ソリッドモデル(点・線・面・体積)

境界表現/CSG表現 → CAD・CGへ利用

多面体メッシュ表現 → CAEへ利用



3D モデルの応用先

3D モデル表現

- 点群
- ワイヤフレームモデル
- サーフェスモデル
- ソリッドモデル

3D モデリング

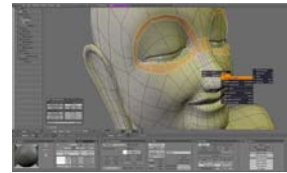
- 3D スキャナ
- CGソフト
- CADソフト
- 画像領域分割

まとめ

モデリング

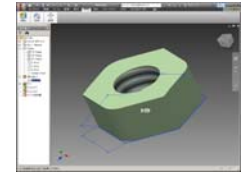
モデリング：3Dモデルを構築する作業

◆ CGソフトウェア



Blender www.blender.org/

◆ CADソフトウェア



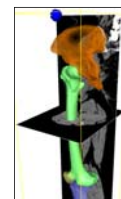
Autodesk Inventor

◆ 3Dスキャナ



Microsoft Kinect

◆ 画像領域分割

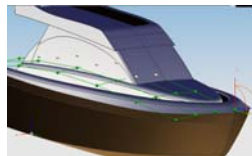


[Ijiri et al.
Eurographics 2013]

モデリング > 3D CG ソフト 1/2 Demo : Blender

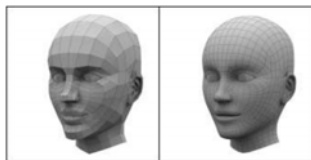
MAYA, 3ds Max, Light wave, **Blender**, Sketch Up, zbrush, Metasquoia, Clip Studio Modeler, etc...

ポリゴンメッシュを構築
様々な自由曲面デザインツールが存在



パラメトリック曲面

[wikipediaよりビデオ](#)



ポリゴン+再分割曲面

[\[Online Design textureよりビデオ\]](#)



Digital sculpting

[Wikipediaよりビデオ](#)

モデリング > 3D CG ソフト 2/2 Meshlab

製造を考慮せずに複雑な自由曲面を生成できる
寸法・Watertight性は厳密でないことが多い



© wikipedia

※ 最近はビジュアルシミュレーションの要求から、寸法に注意したモデリングも必要に
ビジュアルシミュレーション: リアルな映像制作のためのシミュレーション(水・火・煙)

モデリング > CADソフト 1/2

Autodesk Inventor, AutoCAD, CATIA, PTC Creo Parametric, CADMEISTER, ICAD/SX, etc...

CAD (ソリッド) モデルを構築

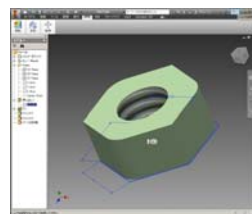
拘束を活用しモデリング

- 長さ・角度・半径を指定
- 拘束 (接・直交・平行) を指定

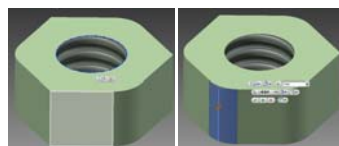
フィーチャーを活用しモデリング

- “製造的”な意味を持つ部分形状
- 押し出し・穴開け・面取りetc...

自由曲面(CGキャラ・顔)は扱いにくい



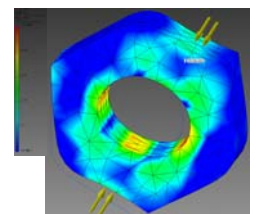
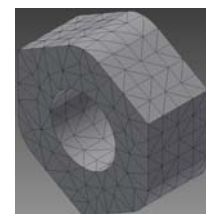
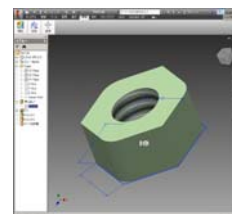
Autodesk Inventor



フィレット構築

モデリング > CADソフト 2/2

一部のCADソフトはCAD → 多面体メッシュ変換をサポート
さらに構造解析までサポートしているものも



By Autodesk Inventor

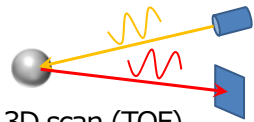
モデリング> 3D スキャナ

KinectTest.exe



Structure from motion

複数写真からその撮影位置を推測し、さらに画素の深さ値を推定



3D scan (TOF)

Time of flight方式。物体に光を投影し返ってきた光との位相差から距離を推測 (Kinect2はこっち)



Kinect ©Microsoft

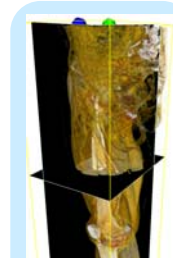


© VIUscan

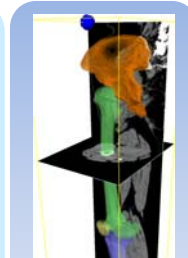
3D scan (Structured light)

Slitなどのパターンを投影しその変形を計測することで表面形状を得る

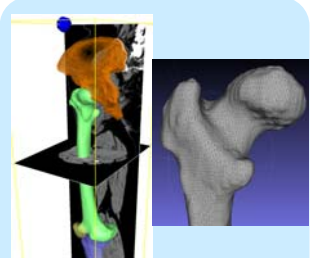
モデリング> 画像領域分割



1) 画像撮影
CT/MRI/顕微鏡



2) 画像領域分割
ImageJ (free: 後期演習)
IMARIS (expensive)
GIMP (2Dのみ free)
VoTracer (理研)
VCAT1.5 (理研)



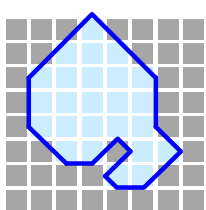
3-1) 三角形ポリゴンモデル構築
画像処理ソフトで行えることが多い
Marching Cubes法
3-2) 四面体メッシュ構築
特殊なソフトが必要

CT画像は理研・画像情報処理研究チームより

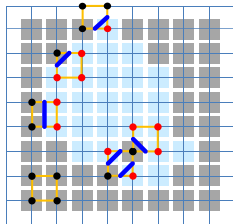
モデリング> 画像領域分割 > Marching Cubes法

ラベル画像(ボクセルデータ)をポリゴンメッシュに変換する
スカラー場の等値面をポリゴンメッシュとして抽出する

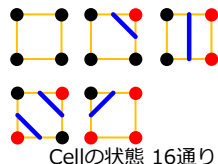
2Dの場合



入力: ラベル画像

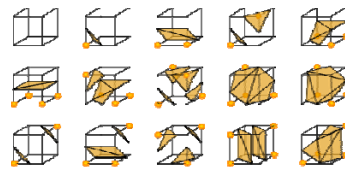


1. グリッドを構築
2. 4 頂点からなる『cell』を走査
頂点の内外状態に応じて
線分を配置



Cellの状態 16通り

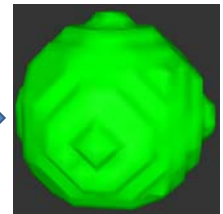
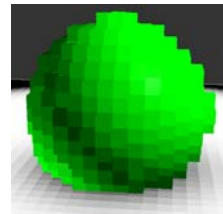
モデリング> 画像領域分割 > Marching Cubes法



Cellの状態 256通り

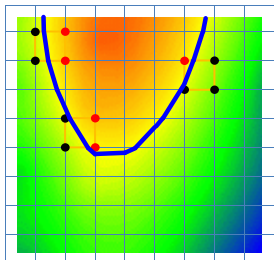
3Dの場合

1. グリッドを構築
2. 8 頂点からなる『cell』を走査
→ 頂点の内外状態に応じて
線分を配置

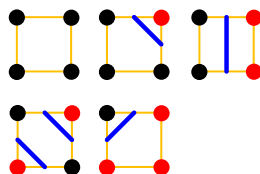


モデリング> 画像領域分割 > Marching Cubes法

2Dスカラー場の等値面を求める場合



1. グリッドを構築
2. 4 頂点からなる『cell』を走査
→ 頂点の内外状態に応じて線分を配置



Cellの状態 16通り
頂点の値に応じて線分の位置を調整
→ Sub-pixel levelの境界が得られる

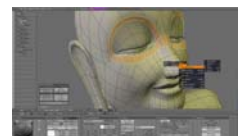
モデリング > まとめ



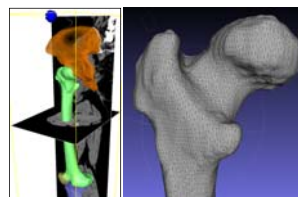
Microsoft Kinect

3Dモデル構築方法はだまかに以下の通り

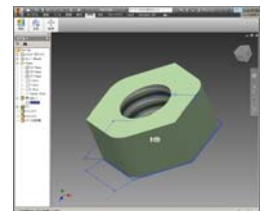
- CG ソフトウェア
- CADソフトウェア
- 3D スキャナ
- 画像領域分割 ← 生体シミュレーション



Blender www.blender.org/



理研・画像情報処理研究チーム



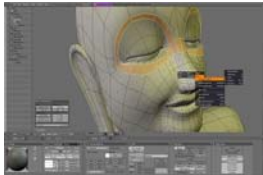
Autodesk Inventor

モデリング > CG と CAD の違い

3D CG ソフト

『絵を作るため』に3Dモデル構築
絵を描くように**自由**にモデリング
→スカルプトツール

曲線：画素より細かな情報は不要
→ ポリライン



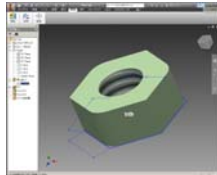
Blender

※ ただし、リアリティはモデルの正確性に起因する部分も大きいのでCGソフトでもある程度の精度は重視される

3D CAD ソフト

『機械が加工する』ためにモデル構築
寸法図を書くように**厳密な**モデリング
→拘束・フィーチャーツール

曲線：加工機械の精度までは必要
→数式で表現



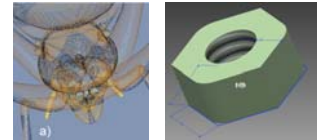
Autodesk Inventor

まとめ

3D モデルの応用先

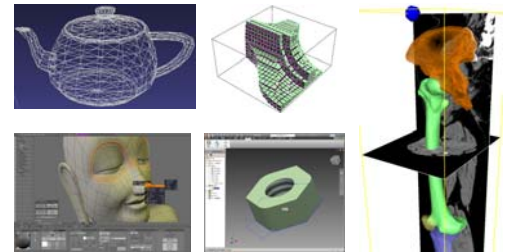
3D モデル表現

- 点群
- ワイヤフレームモデル
- サーフェスモデル
- ソリッドモデル



3D モデリング

- 3D スキャナ
- CGソフト
- CADソフト
- 画像領域分割



まとめ

レポート課題

提出方法：

学籍番号・氏名を記載し、件名を『BME 2015』とする

takashi.ijiri@riken.jpへメールにて送付

課題内容：

1) 次のURLにあるCT画像から三次元ポリゴンモデルを生成せよ

url: www.ms.is.ritsumei.ac.jp/profile/staff/ijiri/classes/tomato.zip

レポートには、用いたソフトウェアと手法を記載せよ

レポートには、三次元モデルをレンダリングした画像を添付せよ

※ ただし、実際にソフトウェアを利用して三次元モデル化が困難な場合は、用いるべき手法を列挙するのみでもかまわない

※ ヒント、フリーソフトがあります『Image J and Marching cubes plug in』

2) 講義の感想を(200文字以内で)述べよ

内容の難易度は適切か？内容には興味が持てたか？など

特に、取り上げてほしい内容などを書いてもらえると参考になります

課題のヒント

- Fijiを利用するなら…
- Fiji (Image J) をダウンロード & インストール
- File > import > Image sequenceより画像をロード
- Plugins > 3D viewerより3次元モード起動
- 3D viewerの Edit > Adjust thresholdより良い感じの閾値を選択
- 3D viewerの Add > From Imageより 先ほどの閾値を用いてSurface 生成
- 3D viewerの File > exportよりwave front objを出力できる