

**画像・映像情報処理(メディア創成) 前半: 工藤博幸  
後半: 藤澤 誠**

**講義を始めるにあたって**

1. 教科書は使用しない。スライド、板書、プリントを中心とする。  
(講義の内容と合致した参考書をテーマごとに紹介)
2. 画像処理の全体像が分かるように広く浅く説明する。
  - ・画像処理の概要と基礎事項(工藤)
  - ・劣化画像の画質改善処理(工藤)
  - ・画像の認識、理解(工藤、藤澤)
  - ・コンピュータグラフィックスによる画像生成(藤澤)
  - ・動画画像処理とコンピュータビジョン(藤澤)
3. 画像処理のプログラムも書けるように演習を2回程度行う。
  - ・空間フィルタを用いた画像処理(工藤)
  - ・OpenCVを用いた画像処理(藤澤)
4. 関連科目 → 信号とシステム(2年), 情報理論(2年), パターン認識(3年)

**講義の内容と合った参考書**

- (1) 田村秀行編著, 『コンピュータ画像処理』, オーム社, 2002年, 3900円
- (2) 末松良一, 山田宏尚著, 『画像処理工学』, コロナ社, 2000年, 3000円
- (3) 南敏, 中村納著, 『画像工学』, コロナ社, 1989年, 2800円

**プログラムが書いてある本**

- (1) 酒井幸市著, 『デジタル画像処理の基礎と応用』, CQ出版, 2003年, 2900円
- (2) 安居院猛, 長尾智晴著, 『C言語による画像処理入門』, 昭晃堂, 2000年, 3000円

**前半のスライドpdfファイルのダウンロード先**

各自ダウンロードして印刷してください

<http://www.cs.tsukuba.ac.jp/~kudo/japanese.html>

ユーザ認証

ユーザID: ??? 授業時間中に教えます  
パスワード: ???

後半: 藤澤先生から指示

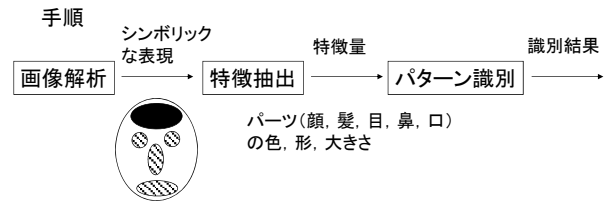
**第1章: 画像処理の概要と基礎事項**

**1. 画像処理の概要**

- ・画像の生成, 伝送, 認識などを目的として計算機で画像を処理
- ・広範囲な技術, 応用分野

**(1) 画像の認識・理解**

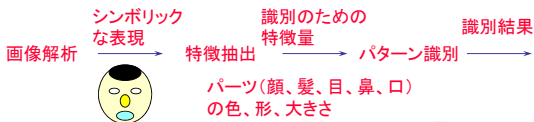
画像に写っているものを計算機に認識させる  
(文字認識, ロボットの環境認識, 個人照合, 医用画像)



**画像の認識・理解**

画像に写っているものをコンピュータに認識させる技術  
(文字認識, ロボットの環境認識, 個人照合, 医用画像認識)

**画像認識の手順**

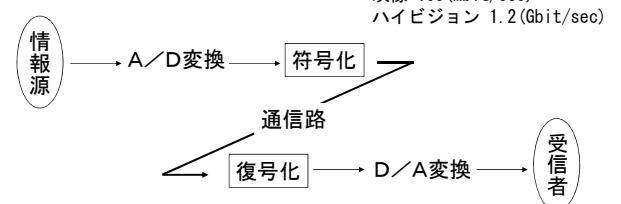


**画像解析(顔画像から構造物を抽出)**



**(2) 画像の符号化, 圧縮**

**画像のデジタル通信のモデル**



**符号化の目的**

- データの圧縮 → 通信を効率的に(特に重要)
- 通信路での誤りが訂正できる形式に変換

**(3) 医用イメージング**

### メディア符号化

音声・画像通信のモデル

音声: 64キロ bit/sec  
映像: 100メガ bit/sec  
ハイビジョン: 1.2ギガ bit/sec

情報源 → A/D変換 → 符号化 → 通信路 → 復号化 → D/A変換 → 受信者

メディア符号化の目的

- (1) 通信路の効率的使用のためデータを圧縮する
- (2) 信頼性向上のため通信路の誤りが訂正可能な形式にする

JPEG 1/200      JPEG 1/150      JPEG 1/100

### 計算機の利用による医療の変化

(a) 計算イメージング(CT, MRI, 超音波)の出現 (1973年)

計測で得たデータに計算を施して断層像を合成

放射線 → 計算 → デジタル画像  
形態, 機能情報

(b) 計算機による医用画像処理技術の発展 (1990年代)

	診断のメディア	診断, 治療の方法
CT発明前	レントゲン写真, 心電図, 脳波	医師
CT発明後	上記+X線CT, MRI(磁気共鳴イメージング), エミッションCT, 超音波	
将来		計算機支援, 自動診断

### 生体イメージングと計算機支援医療

コンピュータの利用による医療の変化

- ・計算イメージング(CT, MRI, 超音波)の出現(1973年)  
(計測で得たデータに計算を施して画像を合成する)

放射線 → 計算 → 断層像

・コンピュータによる医用メディア処理技術の発展(1990年代)

	診断に利用されるメディア	診断・治療の方法
CTの発明以前	レントゲン写真, 心電図, 脳波	医師
CTの発明以降	上記+X線CT, MRI(磁気共鳴イメージング), エミッションCT, 超音波	
将来		計算機支援 完全自動化

(4) 画像の画質改善

撮影やデジタル化の過程で生じた劣化を改善  
(低コントラスト, ぼけ, 雑音, 幾何学的歪)

手順

入力  $f(x, y)$  → 濃度値, 座標に変換 → 出力  $g(x, y)$

(5) 図面, 文書, 印刷画像処理

メディアに固有の画像処理

例

- ・中間調化(ハーフトニング)
- ・文字領域と図形領域の分離

### 劣化画像の品質改善

画像の劣化要因を取り除き人間に見やすい画像に改善

雑音      ぼやけ      解像度低下      コントラスト低下

劣化のモデル  $\vec{g} = A\vec{f} + \vec{n}$  に基づき  $\vec{g}$  から  $\vec{f}$  を推定  
( $f$ : 原画像,  $g$ : 劣化画像,  $A$ : 劣化を表す演算子,  $n$ : 雑音)

低解像度画像 → 改善 → 高解像度画像

### 2. 画像処理の周辺分野

(1) コンピュータビジョン(CV)

人間の奥行き情報知覚機能をコンピュータで実現  
(2次元画像から3次元物体の形状を復元)

3次元シーン  $\vec{x}$  → カメラによる撮影 → 2次元画像  $\vec{y}$

$\vec{y}$  から  $\vec{x}$  を推定して奥行き情報(形状)回復

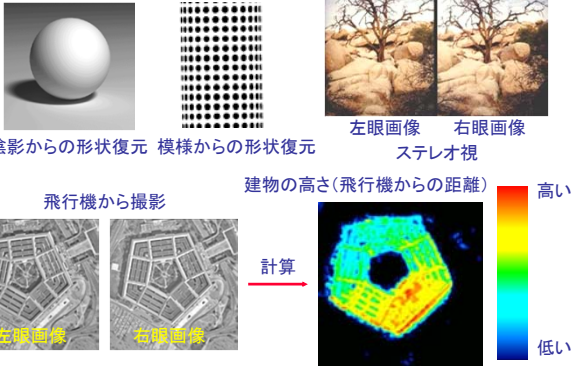
“Shape from X”, 視覚の逆問題と呼ばれる

例

問題	入力データ	計算される量	原理
両眼ステレオ	ステレオ画像	奥行き	三角測量
陰影からの形状復元	濃淡画像	面の方向	Lamberの法則

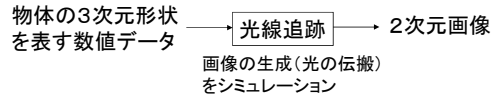
# コンピュータ・ビジョン(CV)

2次元画像から3次元空間(奥行き情報)を知覚する機能の実現



# (2)コンピュータグラフィックス(CG)

コンピュータで人工的に2次元画像を合成



- 分野
- (a) 3次元形状のモデリング(幾何形状, 自然物)
  - (b) 光線追跡法の研究

# コンピュータ・グラフィックス(CG)

画像の生成過程をコンピュータでシミュレーションして画像合成



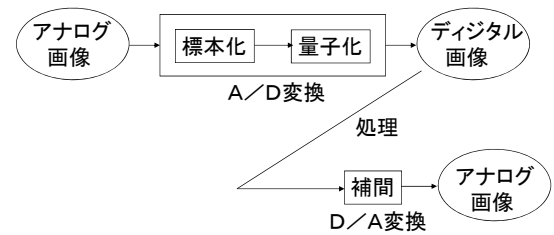
代表的な画像の作り方

- ・レイトレーシング(対象物が球, 直方体などの幾何学的形状の組み合わせ)
- ・ボリュウムレンダリング(対象物がボクセルデータとして表現)



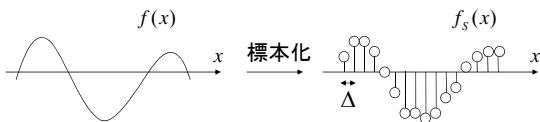
# 3. 画像のデジタル化

デジタル画像処理の手順



# 3. 1 1次元の標本化

信号  $f(x)$  の一定間隔  $\Delta$  おきの値を取り出す



標本化間隔  $\Delta$  の選び方 → 原信号が標本値から再生できるように

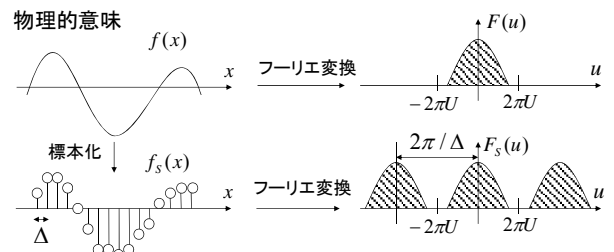
標本化定理 角周波数  
 $f(x) \leftrightarrow F(u)$  フーリエ変換対

$$F(u) \text{ は } U \text{ 以上の周波数成分を含まない } (F(u) = 0 \text{ for } |u| \geq 2\pi U)$$

$$\Rightarrow f(x) = \sum_{k=-\infty}^{\infty} f\left(\frac{k}{2U}\right) h\left(x - \frac{k}{2U}\right) \quad h(x) = \frac{\sin(2\pi Ux)}{2\pi Ux} \quad \text{式(1)}$$

標本値 標本化関数

$\Delta \leq \frac{1}{2U}$  とすれば  $f_s(x)$  から  $f(x)$  が正しく再生可能



標本化後の  $F_s(u)$  は標本化前の  $F(u)$  の繰り返し  
 $F_s(u)$  の一周期分だけフィルタで取り出す

$$F(u) = F_s(u)H(u) \quad H(u) = \begin{cases} 1 & |u| \leq 2\pi U \\ 0 & |u| > 2\pi U \end{cases} \quad \text{フィルタ 式(2)}$$

式(2)のフーリエ逆変換をとると式(1)になる

標本化間隔  $\Delta$  が大きい場合

スペクトルに重なり  
→  $H(u)$  で取り出せない

エリアシング(折り返し)誤差を生じる

$f(x)$  が帯域制限されていない場合の標本化

前置フィルタ →  $f'(x)$  → 標本化 →  $f'_s(x)$

$1/(2\Delta)$  以上の  
周波数成分カット

### 3.2 2次元の標本化

図1

(a) 正方形格子 (b) 正三角形格子 (c) 正六角形格子

正方形格子が最も良く使われている

標本化定理の拡張(正方形格子の場合)

次ページの図参照

$F_S(u, v)$  の1周期分をフィルタで取り出す

$$F(u, v) = F_S(u, v)H(u, v) \quad H(u, v) = \begin{cases} 1 & |u| \leq 2\pi U \text{ and } |v| \leq 2\pi V \\ 0 & \text{otherwise} \end{cases} \quad \text{式(3)}$$

フーリエ変換

フーリエ変換

標本化

$F(u, v)$  の繰り返し

$\Delta \leq \min(\frac{1}{2U}, \frac{1}{2V})$  ならば  $f_S(x, y)$  から  $f(x, y)$  の再生可能

式(3)のフーリエ逆変換をとると

$$f(x, y) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f(\frac{k}{2U}, \frac{l}{2V}) h(x - \frac{k}{2U}) h(y - \frac{l}{2V}) \quad \text{式(4)}$$

標本化関数

図2

### 3.3 量子化

アナログ濃度値  $f(k\Delta, l\Delta)$  を整数値に変換

量子化レベル数の選択

量子化ビット	2	5	8
レベル数	4	32	256

6ビット以上 → 人間の目に不自然に感じない

図4

### 量子化関数の選択

図3

線形  
exp  
log, exp → 非線形

### 量子化誤差

$$e = \sum_k \sum_l (Q[f(k\Delta, l\Delta)] - f(k\Delta, l\Delta))^2$$

量子化演算

### 4. 画像の直交変換

順変換

$$\vec{T}_f = \begin{bmatrix} T_f(1,1) \\ \vdots \\ T_f(N_1, N_2) \end{bmatrix} = \begin{bmatrix} N \times N \\ A \end{bmatrix} \begin{bmatrix} f(1,1) \\ \vdots \\ f(N_1, N_2) \end{bmatrix} \quad \text{画像ベクトル} \quad \vec{f} \quad \text{式(5)}$$

$A$  がユニタリ行列 ( $A^{-1} = A^T$ ) のとき → 直交変換

$$\vec{T}_f = A\vec{f}, \quad \vec{f} = A^T\vec{T}_f \quad \text{式(6)}$$

順  
逆

水平方向と垂直方向に分離可能な直交変換の一般形

$$T_f(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f(n_1, n_2) a_R(n_1; k_1) a_C(n_2; k_2) \quad \text{式(7)}$$

周波数  $f(n_1, n_2) = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} T_f(k_1, k_2) b_R(n_1; k_1) b_C(n_2; k_2)$  基底

板書

直交変換  $a(n_1; k_1; n_2; k_2)$   
 $b(n_1; k_1; n_2; k_2)$

1次元変換の繰り返しで計算可能

1次元変換のみを考えて十分

(1) 離散フーリエ変換 (DFT)

$$T_f(k) = \sum_{n=0}^{N-1} f(n) e^{-j \frac{2\pi kn}{N}}$$

複素数 式(8)

連続フーリエ変換の近似

板書

フーリエ  $F(\omega) = \int_0^{N-1} f(x) e^{-j\omega x} dx$   
 $F(\omega) \approx \sum_{n=0}^{N-1} f(n) e^{-j\omega n}$   
 $\omega = \frac{2\pi k}{N}$

フィルタ

フィルタ

$$f(x) \xrightarrow{F} F(\omega) \xrightarrow{\times H(\omega)} G(\omega) \xrightarrow{F^{-1}} g(x)$$

板書

JPEG

エネルギー集中

DC

$\approx 0$

$T_f(k_1, k_2)$

用途

- 特徴抽出(方向性, 周波数分布) →
- フィルタリング(鮮鋭化, 雑音除去)
- CTの画像再構成

高速計算法(FFT)が存在する

(2) 離散コサイン変換(DCT)

順  $T_f(k) = \alpha(k) \sum_{n=0}^{N-1} f(n) \cos\left(\frac{\pi(2n+1)k}{2N}\right)$   $\alpha(0) = \sqrt{\frac{1}{N}}$  式(9)

逆  $f(n) = \sum_{k=0}^{N-1} \alpha(k) T_f(k) \cos\left(\frac{\pi(2n+1)k}{2N}\right)$   $\alpha(k) = \sqrt{\frac{2}{N}} (1 \leq k \leq N-1)$

用途

- 変換符号化における信号の冗長さの削減(JPEG)

図5

(3) アダマール変換

データ数  $N = 2^l$

変換の行列表現

$$N \begin{bmatrix} T_f(k) \end{bmatrix} = \begin{bmatrix} H_N \end{bmatrix} \begin{bmatrix} f(n) \end{bmatrix} N$$

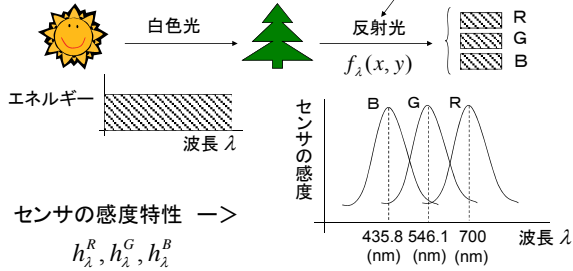
式(10)

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad H_{2n} = \frac{1}{\sqrt{2}} \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix} (n=2,4,8,\dots)$$

要素が 1 or -1 → 加減算のみで実現可能

## 5. カラー画像とマルチスペクトル画像

カラー画像の生成過程



カラー画像

$$f^R(x, y) = \int f_\lambda(x, y) h_\lambda^R d\lambda$$

$$f^G(x, y) = \int f_\lambda(x, y) h_\lambda^G d\lambda$$

$$f^B(x, y) = \int f_\lambda(x, y) h_\lambda^B d\lambda$$

図6

式(11)

一般化

波長に依存する信号  $f_\lambda(x, y)$  を複数の異なる感度特性を持つセンサで計測してできる画像の組

$\rightarrow$  マルチスペクトル画像

例 リモートセンシング画像

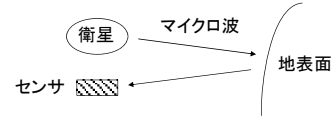


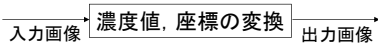
図7

## 第2章: 画像の画質改善処理

画像に含まれる雑音や歪を取り除く  
(コントラストの強調, 雑音除去, ぼやけ補正, 幾何学的歪の補正)

目的

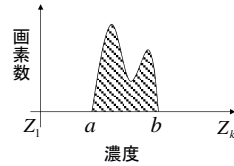
- (1) 画像を人間に見やすくする
- (2) 特徴抽出や認識が容易に行えるように  $\rightarrow$  前処理



### 1. 濃度変換と強調

#### 1.1 コントラストの強調

コントラストの低下  $\rightarrow$  濃度ヒストグラムの範囲が狭い



- ・記録装置のダイナミックレンジ不足
- ・露出不足
- ・デジタル化の際の濃度設定不適切

#### (1) 線形な濃度変換

次式で引き伸ばしと平行移動

$$Z' = \frac{Z_k - Z_1}{b - a} (Z - a) + Z_1$$

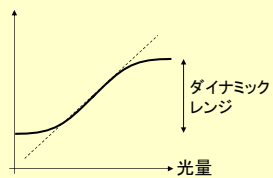
変換後  $\quad$   $\quad$   $\quad$  変換前

式(1)

板書

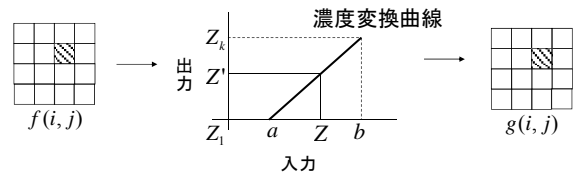
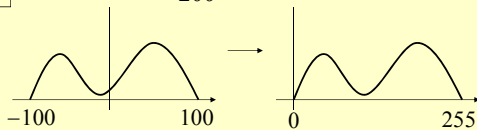
センサーで記録される信号強度

ダイナミックレンジ



負の画像

$$Z' = \frac{255}{200} (Z + 100)$$



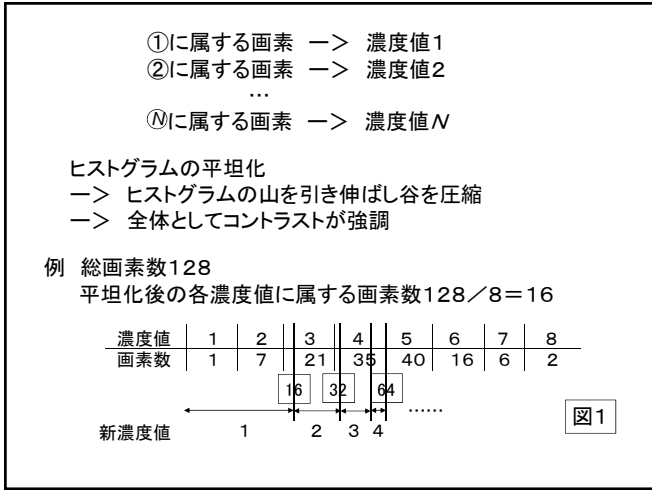
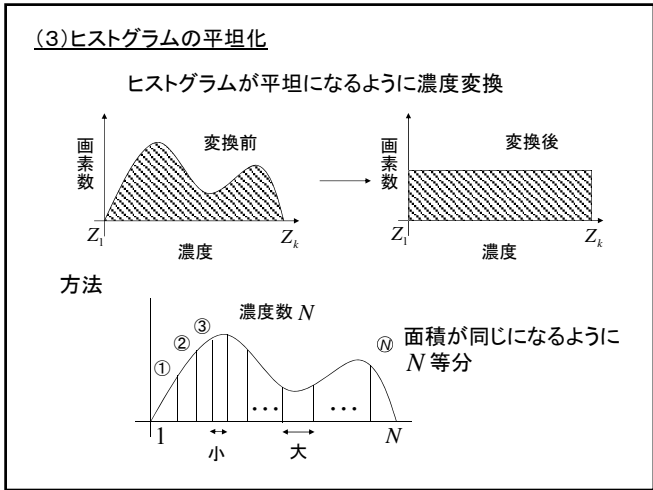
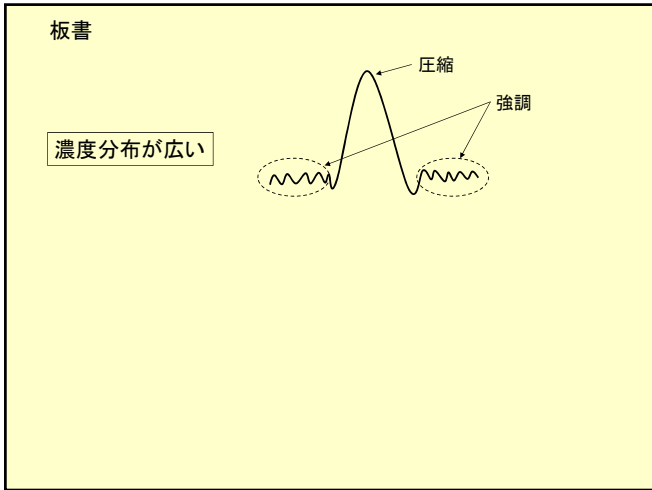
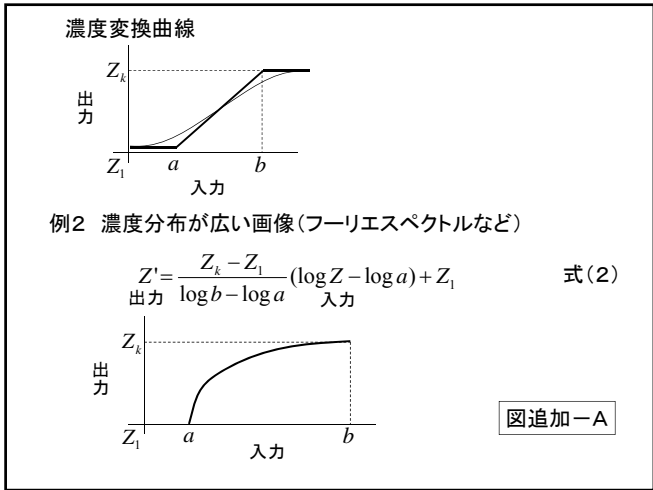
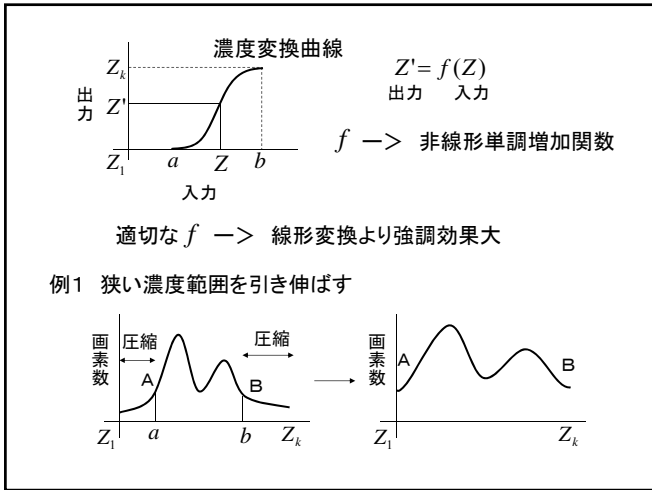
濃度変換曲線  $\rightarrow$  入力の一価関数

別の応用例

負の濃度値を持つ画像を正の濃度値しか扱えない表示装置で見る

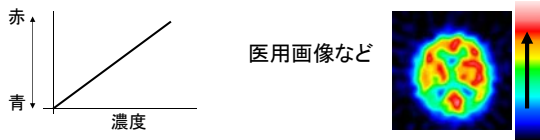
#### (2) 非線形な濃度変換

濃度変換曲線が非線形



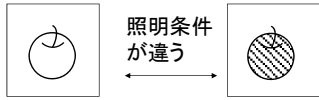
(4) 擬似カラー表示

濃度値を色に対応させて変換 → 人間に見やすくなる



(5) 濃度の正規化

異なる条件で撮影した画像を比較(画像の認識)



ヒストグラムが一定の形になるように濃度変換

平均値  $m_0$ , 標準偏差  $\sigma_0$

平均  $m$ , 標準偏差  $\sigma$  のヒストグラムを平均  $m_0$ , 標準偏差  $\sigma_0$  に変換

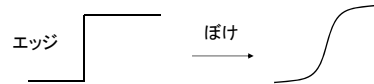
$$Z' = \frac{\sigma_0}{\sigma} (Z - m) + m_0$$

出力                      入力

1.2 濃度の鮮鋭化

“ぼけ”を取り除き画像を鮮明なものに

“ぼけ”の原因 → 高周波が弱められている



考え方 → 高周波成分の強調

(1) 空間フィルタによる方法

ラプラシアン

出力画像 = 原画像 - 2次微分画像

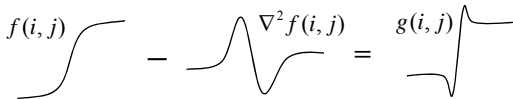
出力

$$g(i, j) = f(i, j) - \nabla^2 f(i, j) \quad \text{式(3)}$$

$$= 5f(i, j) - [f(i+1, j) + f(i-1, j) + f(i, j+1) + f(i, j-1)]$$

ラプラシアン(2次微分)

$$\nabla^2 f(i, j) = f(i+1, j) + f(i-1, j) + f(i, j+1) + f(i, j-1) - 4f(i, j) \quad \text{式(4)}$$

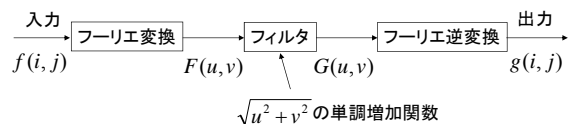


板書

2次微分  $\nabla^2 f(x, y) = \frac{\partial^2}{\partial x^2} f(x, y) + \frac{\partial^2}{\partial y^2} f(x, y)$

濃度の傾斜が急峻に  
エッジの上端, 下端でオーバーシュート, アンダーシュート  
→ 人間に見やすくなる

(2) 周波数フィルタによる方法



問題点

画像成分とともに雑音も強調する → やりすぎない

図2

鮮鋭化(空間フィルタ)





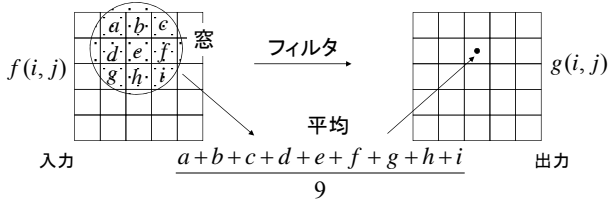
## 2. 平滑化と雑音除去

ランダム雑音を除去する

雑音が統計的にモデル化可 → 画像復元(フィルタの設計)  
 雑音が統計的にモデル化不可 → 平滑化(経験的手法)

### 2.1 平均値フィルタ

出力画像の点  $(i, j)$  の濃度 = 入力画像の点  $(i, j)$  近傍の平均濃度



窓サイズ  $(2n+1) \times (2n+1)$  のとき

$$g(i, j) = \sum_{k=-n}^n \sum_{l=-n}^n \frac{1}{(2n+1)^2} f(i+k, j+l) \quad \text{式(5)}$$

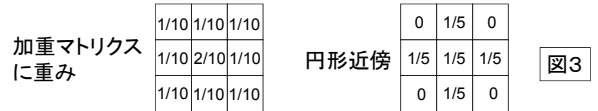
窓サイズの決定

$n \rightarrow$  大のとき 雑音除去効果大, エッジがぼける

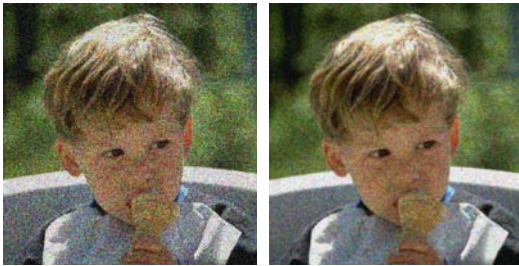
$n \rightarrow$  小のとき 雑音除去効果小, エッジ保存

$n$  は大きすぎず, 小さすぎない値を選択

平均値フィルタの変形



### ・雑音除去



入力画像

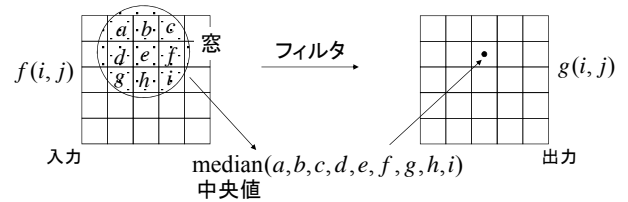
フィルタの出力

画像は低周波数成分を多く含み雑音は高周波数成分を多く含むことに着目して, 低周波数成分のみを通過するフィルタリングを行っている

### 2.2 中央値(メディアン)フィルタ

エッジのぼけを軽減する雑音除去

出力画像の点  $(i, j)$  の濃度 = 入力画像の点  $(i, j)$  近傍濃度の中央値



非線形 →  $H(f_1 + f_2) = Hf_1 + Hf_2$  が成立しない  
 フィルタ演算 (平均値フィルタは線形)

### 板書

線形、非線形

$$H(f + n) = Hf + Hn$$

### 例 1次元で窓サイズ3

入力 ... 0, 1, 0, 1, 6, 5, 6, 6, ...  
 平均値 ... 0, 0, 1, 2, 4, 6, 6, 6, ...  
 中央値 ... 0, 0, 1, 1, 5, 6, 6, 6, ...

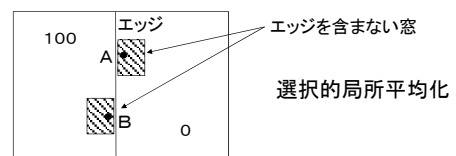
### 2.3 エッジを保ったスムージング(Edge Preserving Smoothing)

平均値フィルタにおけるエッジのぼけ

→ 窓をエッジを含むように選ぶのが原因

解決策

窓をエッジを含まない局所領域に選ぶ



(1) エッジ検出を行う方法

エッジ検出を行いエッジを含まない窓を選ぶ

(2) エッジ検出を行わない方法

9つの窓の分散と平均濃度を計算

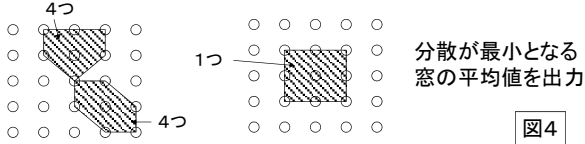


図4

様々な平滑化法の比較

- (1) 雑音除去能力 → ほぼ同じ → 悪
- (2) エッジ保存 エッジ保存 > 中央値 > 平均値
- (3) 計算量 エッジ保存 > 中央値 > 平均値 → 少

図5

板書

平均と分散

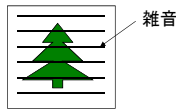
$$m = \frac{1}{N_W} \sum_{(i,j) \in W} f(i,j)$$

$$\sigma^2 = \frac{1}{N_W} \sum_{(i,j) \in W} (f(i,j) - m)^2$$

2.4 孤立雑音の除去

規則性のある雑音や孤立点の除去 → 雑音の性質を利用

(1) 細い線状の周期性雑音

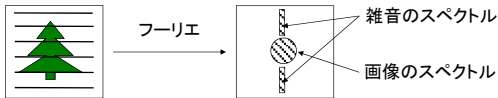


方法1

線の両側の画素から線上の画素の濃度を補間

方法2

フーリエ変換 → 周期性雑音のスペクトル除去 → フーリエ逆変換



(2) 孤立点(ごま塩雑音)の除去

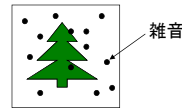
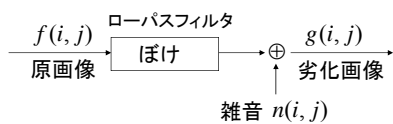


図6

孤立点を検出(周囲に類似した画素値が少ない)して周囲の平均値に置き換える

3. 画像の復元 (Image Restoration)

ぼけ+雑音で劣化した画像の改善



劣化の過程が数式モデルで表現できる

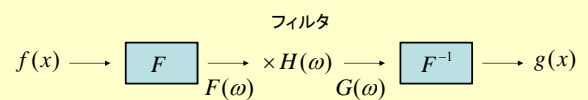
$$g(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x-\alpha, y-\beta) f(\alpha, \beta) d\alpha d\beta + n(x,y) \quad \text{式(6)}$$

$h(x,y)$  → 点拡がり関数 (座標原点にある1つの明るい点がどのようにぼけるか)

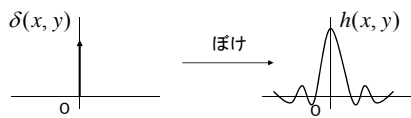
板書

フィルタと畳み込み積分

1次元信号



$$g(x) = \int_{-\infty}^{\infty} h(x-\alpha) f(\alpha) d\alpha \quad \text{畳み込み積分}$$



### 3.1 逆フィルタ

雑音がない

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x - \alpha, y - \beta) f(\alpha, \beta) d\alpha d\beta \quad \text{式(7)}$$

両辺のフーリエ変換

$$G(u, v) = H(u, v)F(u, v) \quad \text{式(8)}$$

画像復元

$$\hat{F}(u, v) = G(u, v) / H(u, v) \quad \text{式(9)}$$

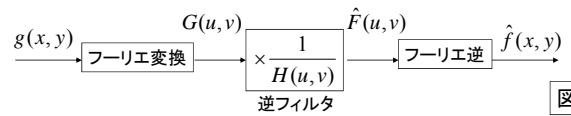


図7

問題点 → 雑音の拡大

雑音のフーリエ変換

$$\hat{F}(u, v) = \frac{G(u, v) - N(u, v)}{H(u, v)} = F(u, v) - \frac{N(u, v)}{H(u, v)} \quad \text{式(10)}$$

雑音の影響

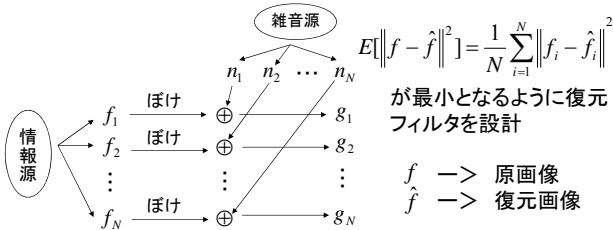
$H(u, v) \approx 0$  となる周波数があると第2項は大きくなる

### 3.2 ウィーナー・フィルタ(最小2乗フィルタ)

逆フィルタの雑音増大を抑える

考え方

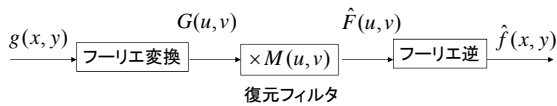
ウィーナーの雑音を含む信号の推定理論



$$E[\|f - \hat{f}\|^2] = \frac{1}{N} \sum_{i=1}^N \|f_i - \hat{f}_i\|^2$$

が最小となるように復元フィルタを設計

$f$  → 原画像  
 $\hat{f}$  → 復元画像



$$M(u, v) = \frac{1}{H(u, v) \sqrt{|H(u, v)|^2 + [P_{nn}(u, v) / P_{ff}(u, v)]}} \quad \text{式(11)}$$

逆フィルタ

平滑化

$$P_{ff}(u, v) = E[|F(u, v)|^2] \quad \text{画像のパワースペクトル}$$

$$P_{nn}(u, v) = E[|N(u, v)|^2] \quad \text{雑音のパワースペクトル}$$

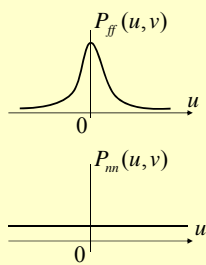
実用上は  $P_{ff}(u, v), P_{nn}(u, v)$  が未知

→  $P_{nn}(u, v) / P_{ff}(u, v)$  を定数で置き換える

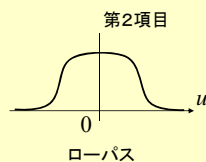
図8

板書

第2項目はローパス



$$\frac{|H(u, v)|^2}{|H(u, v)|^2 + [P_{nn}(u, v) / P_{ff}(u, v)]}$$



・ぼけ補正 (ぼけ画像復元)

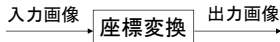


入力画像

フィルタの出力

ぼけは高周波数成分の不足によって生じるため、高周波数成分を補い増強するフィルタリングを行っている

#### 4. 幾何学的変換

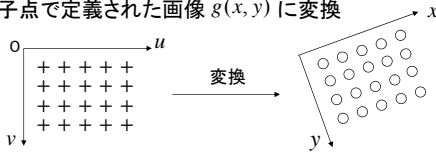


##### 目的

- ・画像の平行移動, 回転, 拡大縮小
- ・レンズや撮像系のもつ幾何学的歪の補正
- ・異なった2種類の画像の重ね合わせ (衛星写真と地図)

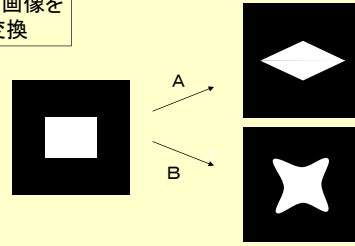
##### 定式化

$(u, v)$ 座標系の格子点で定義された画像  $f(u, v)$  を  $(x, y)$ 座標系の格子点で定義された画像  $g(x, y)$  に変換



#### 板書

長方形の画像をアフィン変換



$(u, v)$ と $(x, y)$ の関係

多項式

$$u = p(x, y) = \sum_i \sum_j a_{ij} x^i y^j, \quad v = q(x, y) = \sum_i \sum_j b_{ij} x^i y^j \quad \text{式(12)}$$

(1) アフィン変換(線形変換) → 平行移動, 回転, 拡大縮小

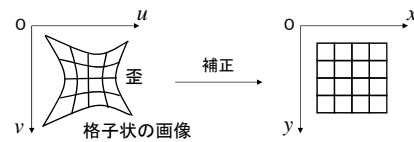
式(12)で2次以上の項がない

$$u = a_{10}x + a_{01}y + a_{00}, \quad v = b_{10}x + b_{01}y + b_{00} \quad \text{式(13)}$$

$$\begin{cases} a_{10} = b_{10} = 1, a_{01} = b_{01} = 0 & (a_{00}, b_{00}) \text{の平行移動} \\ a_{00} = b_{00} = 0, a_{10} = b_{10} = \cos \theta, a_{01} = -b_{10} = -\sin \theta & \text{回転 (原点中心)} \\ a_{01} = b_{01} = 0, a_{00} = b_{00} = 0 & \text{拡大縮小} \end{cases}$$

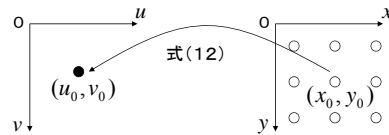
(2) 非線形変換

式(12)で2次以上の項がある → 幾何学的歪の補正

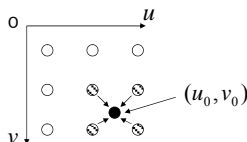


##### 手順

(1)  $(x, y)$ 座標系の格子点  $(x_0, y_0)$ に対応する  $(u, v)$ 座標系の座標  $(u_0, v_0)$ を計算(式(12))



(2)  $(u_0, v_0)$ の濃度を周囲の格子点上の濃度から補間



(3)  $g(x_0, y_0)$ の値を補間した  $f(u_0, v_0)$ の値とする

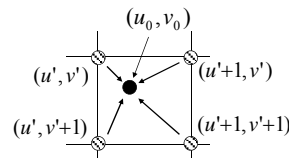
(4)  $(x, y)$ 座標系の全格子点について(1)~(3)の処理を行う

濃度補間の方法

(1) 最近傍法

$(u_0, v_0)$ に最も近い格子点の値を用いる

(2) 線形補間法



周囲の4格子点からの距離に応じた重みつき総和

$$\begin{cases} f(u_0, v_0) = f(u', v')(1-\alpha)(1-\beta) + f(u'+1, v')\alpha(1-\beta) \\ \quad + f(u', v'+1)(1-\alpha)\beta + f(u'+1, v'+1)\alpha\beta \quad \text{式(14)} \\ \alpha = u_0 - u', \quad \beta = v_0 - v' \end{cases}$$

(3) 3次補間法

周囲の16個の格子点の値を用いて3次式で補間

格子点上の値

$$f(u_0, v_0) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f(u'_k, v'_l) C(u'_k - u_0) C(v'_l - v_0)$$

重み

$$C(x) = \begin{cases} 1 - 2|x|^2 + |x|^3 & 0 \leq |x| < 1 \\ 4 - 8|x| + 5|x|^2 - |x|^3 & 1 \leq |x| < 2 \\ 0 & |x| \geq 2 \end{cases}$$

式(15)

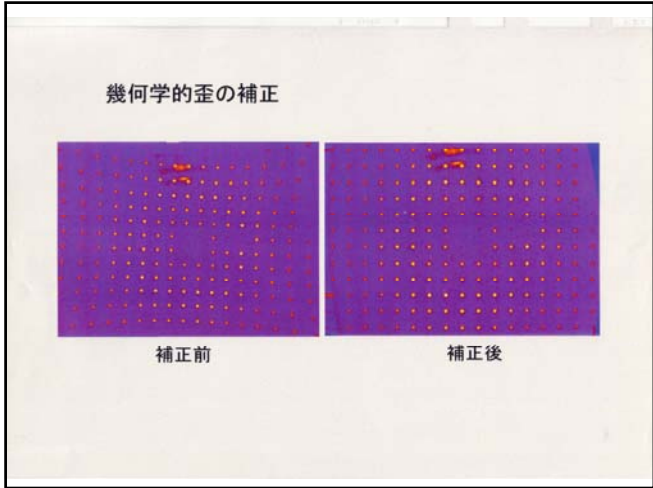
精度の点で理想的な補間 → サンプリング定理によるもの  
(1つの値を補間するのに全標本値を使うので非実用的)

図9

板書

3次補間法

線形補間法



### 第3章: 画像の解析 (Image Analysis)

画像を人間の解釈に近いsymbolicな表現に変換  
→ 画像認識に不可欠な処理

1. 画像認識とは

人間の視覚情報処理の機能を計算機で実現  
(写っているものの色・形・大きさから何が写っているか分かる)

応用例

- 文字認識 (郵便番号, ワープロ, 文書の電子化)
- 画像認識 (より一般の画像を対象) → シーン, 顔画像, 図面

画像認識の一般的手順

次ページの図参照

画像の解析

```

    入力画像 → 前処理 (雑音除去, コントラスト強調) → 領域分割 → 特徴抽出 → 画像のsymbolicな表現
    
```

パターン認識 (知識との照合) → 認識結果

領域分割 → 画像を特徴(色, 模様)が一様な幾つかの領域に分ける

特徴抽出 → 各領域の形状特徴, 濃度特徴, 領域間の相対関係を抽出する

形状特徴 (トポロジー, 面積, 円形度など)  
濃度特徴 (色, 模様など)  
領域間の相対関係 (接続関係, 境界線の長さなど)

板書

領域分割

山 空  
道路

例1 果実画像の認識(りんご, みかん, ばなな)

入力 → 領域分割 → B → 特徴抽出 → symbolicな表現

“赤い色”, “丸い”  
領域Aが1つ

識別 → “この画像はりんご”

知識  
りんご → “赤い”, “丸い”  
みかん → “オレンジ”, “楕円”  
ばなな → “黄色”, “細長い”

ITS (Intelligent Transport System)

例2 複雑な例 → 車両の識別(セダン, トラック, バス)

入力 → 領域分割 → A, B, C, D → 特徴抽出 → symbolicな表現

A ボデー  
B タイヤ  
C タイヤ

グラフの各ノードに形状特徴, 濃度特徴

例3 文字認識

入力 → 領域分割 → A, B → 特徴抽出 (構造解析) → symbolicな表現

端点, 屈折点, 分岐点, 端点

知識  
端, 端, 屈分, 端, 端, 端, ……

識別 → “この文字はア”

画像認識の難しさ  
(1) 安定な画像解析が難しい  
(2) 対象を限定する必要性

図1~3

2. 画像の領域分割(セグメンテーション)と境界線の抽出

- 画像を特徴(色, 模様)が同じ幾つかの領域に分割
- 画像から境界線を抽出

2値化, エッジ検出+エッジ結合, クラスタリング, 領域成長法などの様々な手法がある

2.1 2値化

背景(黒) 256値白黒画像  
対象物(白)  $f(m,n)$  画像の中の1つの対象物を抽出する

原理  
しきい値  $T$  を与え,  $T$  より大きい画素を対象物, それ以外の画素を背景とする

板書

境界線抽出

式(1)

$$B(m,n) = \begin{cases} 0 & f(m,n) \leq T \text{ のとき} \\ 1 & f(m,n) > T \text{ のとき} \end{cases}$$

2値化された画像

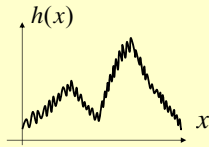
しきい値  $T$  の決め方  
濃度ヒストグラムを用いる

頻度  $h(x)$  背景 対象物 濃度  $x$  2つのピーク

(1) モード法  
濃度ヒストグラムの谷底にしきい値  $T$  を設定

板書

実画像の濃度  
ヒストグラム

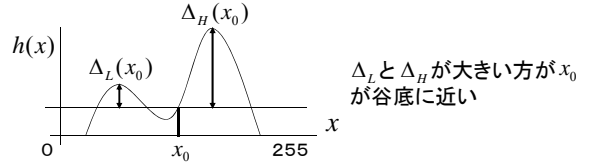


決め方

任意の濃度  $x$  に対して次式を計算

$$\Delta_L(x) = \max_{x'} (h(x') - h(x)) \quad (x' = 0, 1, \dots, x-1) \quad \text{式(2)}$$

$$\Delta_H(x) = \max_{x'} (h(x') - h(x)) \quad (x' = x+1, x+2, \dots, 255)$$

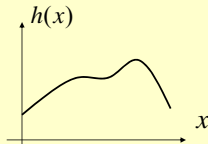


$$T = \max_x \Delta_H \Delta_L \quad \Delta_L \text{と} \Delta_H \text{の積を最大に} \quad \text{式(3)}$$

ヒストグラムの双峰性がはっきりしている時のみ有効

板書

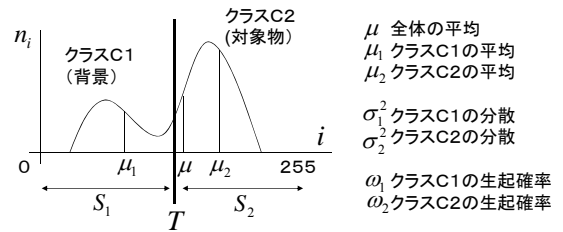
双峰性がはっきりしない  
ヒストグラム



(2) 判別分析法

対象物と背景の濃度分布の分離度が最大となるように  $T$  を決定

$$\frac{\text{背景と対象物の濃度の違い}}{\text{背景の濃度のばらつき} + \text{対象物の濃度のばらつき}}$$



判別分析法の分離度を計算するのに使う統計量

平均

$$\mu = \frac{1}{N} \sum_{i \in S_1 + S_2} i n_i \quad \mu_1 = \frac{1}{N_1} \sum_{i \in S_1} i n_i \quad \mu_2 = \frac{1}{N_2} \sum_{i \in S_2} i n_i$$

分散

$$\sigma_1^2 = \frac{1}{N_1} \sum_{i \in S_1} n_i (i - \mu_1)^2 \quad \sigma_2^2 = \frac{1}{N_2} \sum_{i \in S_2} n_i (i - \mu_2)^2$$

生起確率

$$\omega_1 = \frac{N_1}{N} \quad \omega_2 = \frac{N_2}{N}$$

$$N = \sum_{i \in S_1 + S_2} n_i \quad N_1 = \sum_{i \in S_1} n_i \quad N_2 = \sum_{i \in S_2} n_i$$

全画素数      クラスC1の画素数      クラスC2の画素数

分離度

$$\eta(T) = \frac{\sigma_B^2(T)}{\sigma_W^2(T)} \rightarrow \max \quad \text{式(4)}$$

$\sigma_B^2(T)$  クラス間分散,  $\sigma_W^2(T)$  クラス内分散

$$\sigma_W^2 = \omega_1 \sigma_1^2 + \omega_2 \sigma_2^2 = \frac{1}{N} \left\{ \sum_{i \in S_1} (i - \mu_1)^2 n_i + \sum_{i \in S_2} (i - \mu_2)^2 n_i \right\}$$

クラスC1の濃度のばらつき      クラスC2の濃度のばらつき      式(5)

$$\sigma_B^2 = \omega_1 (\mu_1 - \mu)^2 + \omega_2 (\mu_2 - \mu)^2$$

$$= \frac{1}{N} \left\{ \sum_{i \in S_1} (\mu_1 - \mu)^2 n_i + \sum_{i \in S_2} (\mu_2 - \mu)^2 n_i \right\} \quad \text{式(6)}$$

$N$ : 全画素数

決定方法 図4~6

任意の濃度  $i$  について分離度  $\eta(i)$  を計算

$$T = \max_i \eta(i) \quad \text{式(7)}$$

2.2 エッジ検出による領域分割と境界線の抽出

エッジ → 画像の濃度変化の不連続部  
領域の境界に対応する

板書

エッジ

エッジの種類

実画像の大半      特殊

エッジ検出による領域分割の手順

エッジ結合 → エッジ要素を追跡し途切れなどをつなぎあわせて領域情報に変換する

板書

正四面体

特徴

2値化やクラスタリングと比較して、各領域内の濃度が一様でない場合にも安定

2.2.1 1次微分によるエッジ検出

スロープエッジに対する1次微分

エッジは微分出力の頂点  
エッジの強度は出力の大きさに比例

エッジ検出の手順

しきい値処理 → 雑音によるエッジや認識に不要な弱いエッジを消去

細線化 → エッジを細めて中心線のみを抽出

1次微分の離散近似  
2次元デジタル画像  $f(m,n)$  → 差分で近似



板書

微分の差分近似

$$\frac{df(x)}{dx} = \lim_{\Delta \rightarrow 0} \frac{f(x+\Delta) - f(x)}{\Delta}$$

方法1  $\begin{cases} f_x(m,n) = f(m+1,n) - f(m,n) & x\text{方向の微分} \\ f_y(m,n) = f(m,n+1) - f(m,n) & y\text{方向の微分} \end{cases}$  式(8)

方法2  $\begin{cases} f_x(m,n) = f(m+1,n) - f(m-1,n) & x\text{方向の微分} \\ f_y(m,n) = f(m,n+1) - f(m,n-1) & y\text{方向の微分} \end{cases}$  式(9)

勾配 (gradient) と呼ぶ

1次微分は周波数領域で  $j\omega$  を乗じると等価

→ 空間領域では畳み込み積分で表せる コンボリューション

x方向  $f_x(m,n) = f(m,n) \otimes g_h(m,n)$   
 $= \sum_{m'} \sum_{n'} g_h(m-m', n-n') f(m', n')$  式(10)

y方向  $f_y(m,n) = f(m,n) \otimes g_v(m,n)$

板書

マスク処理の式確認

$\dots + g(1,1)f(m-1,n-1) + g(0,1)f(m,n-1) + g(-1,1)f(m+1,n-1)$  n-1行  
 $+ g(1,0)f(m-1,n) + g(0,0)f(m,n) + g(-1,0)f(m+1,n)$  m行  
 $+ g(1,-1)f(m-1,n+1) + g(0,-1)f(m,n+1) + g(-1,-1)f(m+1,n+1)$  n+1行  
 $\dots$   
-1 1

$g_h(-m,-n), g_v(-m,-n)$  をマスクと呼ぶ

方法1  $\begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix}$   $g_h(-m,-n)$      $\begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix}$   $g_v(-m,-n)$

方法2  $\begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$   $g_h(-m,-n)$      $\begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$   $g_v(-m,-n)$

畳み込み演算で表される画像処理をマスク処理と呼ぶ  
 (エッジ検出の他に平滑化, 鮮鋭化など)

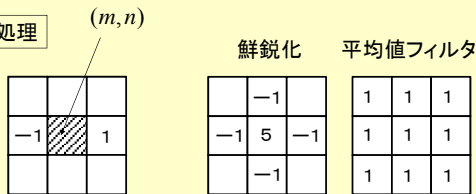
次ページの図でマスク処理を説明

実用上の問題点

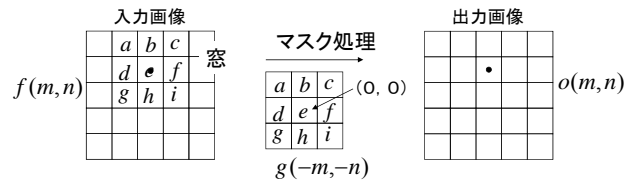
1次微分はハイパスフィルタで雑音増大  
 → 平滑化の効果を入れたマスク

板書

マスク処理



マスク処理の手順



- (1) 出力を求める座標  $(m,n)$  とマスクの原点が一致するようにマスク  $g(-m,-n)$  を入力画像  $f(m,n)$  に重ねる
- (2) 対応する画素の入力画像の値  $f(m,n)$  とマスクの値  $g(-m,-n)$  を乗じて全画素について合計する
- (3) (2) で求めた値を出力画像  $o(m,n)$  の座標  $(m,n)$  に書き込む
- (4) 全画素について(1)~(3)の処理を行う(ラスタ走査の順で)

様々なエッジ検出フィルタ

図7~9

(1)ソーベル(Sobel)フィルタ

微分と直角な方向に平滑化を行うことで雑音抑制

$$g_h(-m,-n) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \begin{matrix} \text{平滑化} \\ \uparrow \\ \text{微分} \end{matrix} \quad g_v(-m,-n) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{matrix} \text{平滑化} \\ \leftarrow \\ \text{微分} \end{matrix}$$

フィルタ出力  $g(m,n) = \sqrt{f_x(m,n)^2 + f_y(m,n)^2}$  式(11)

(2)ロバーツ(Roberts)フィルタ

微分を垂直, 水平方向でなく斜め方向に行うことで雑音抑制

$$g_h(-m,-n) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{matrix} \text{微分} \\ \swarrow \end{matrix} \quad g_v(-m,-n) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{matrix} \text{微分} \\ \searrow \end{matrix}$$

・輪郭線(エッジ)検出



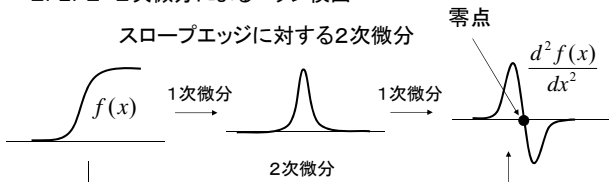
入力画像

フィルタの出力

輪郭部分は平坦部と比較して高周波数成分を多く含むため、高周波成分を強調するフィルタリングを行っている

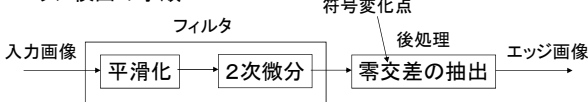
2.2.2 2次微分によるエッジ検出

スロープエッジに対する2次微分



エッジは2次微分の零交差(Zero Crossing)

エッジ検出の手順



平滑化 → 雑音や認識に不要な弱いエッジを除去

特徴

1次微分による手法と比較して後処理が簡単

2次微分の離散近似

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \quad \text{ラプラシアン} \quad \text{式(12)}$$

差分近似すると

$$\nabla^2 f(m,n) = [(f(m+1,n) - f(m,n)) - (f(m,n) - f(m-1,n))] + [(f(m,n+1) - f(m,n)) - (f(m,n) - f(m,n-1))] \quad \text{式(13)}$$

$$= f(m+1,n) + f(m-1,n) + f(m,n+1) + f(m,n-1) - 4f(m,n)$$

フィルタ(マスク処理)の形で表すと

$$\text{出力} \quad g(m,n) = f(m,n) \otimes h(m,n)$$

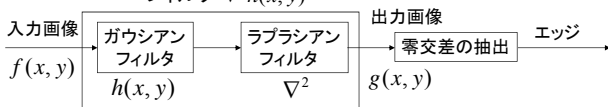
$$h(-m,-n) = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{マスク} \quad \text{式(14)}$$

図10, 11

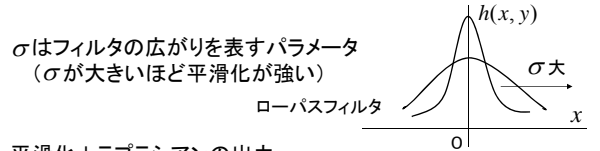
ラプラシアンガウシアン法(Marr & Hildreth)

平滑化フィルタをガウシアンフィルタに選んだもの

フィルタ  $\nabla^2 h(x,y)$



$$\text{ガウシアンフィルタ} \quad h(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \text{式(15)}$$

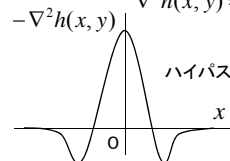


平滑化+ラプラシアン of 出力

$$g(x,y) = \nabla^2 (f(x,y) \otimes h(x,y)) = f(x,y) \otimes (\nabla^2 h(x,y)) \quad \text{式(16)}$$

ラプラシアンガウシアン(平滑化とラプラシアンを1つのフィルタで表したもの)

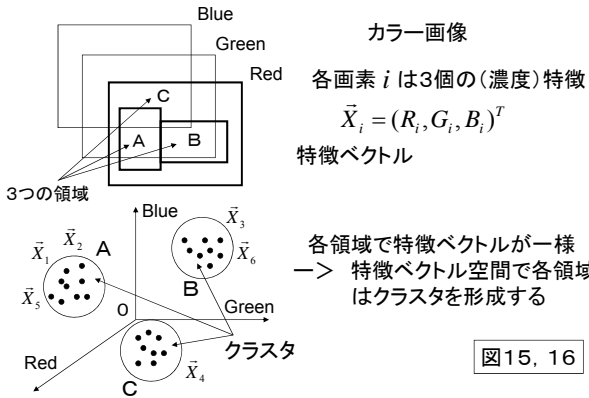
$$-\nabla^2 h(x,y) = -\frac{1}{\sqrt{2\pi\sigma^3}} \left( 2 - \frac{x^2+y^2}{\sigma^2} \right) e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \text{式(17)}$$



人間の視覚情報処理と対応がとくされている

図12~14

### 2.3 クラスタリング (Clustering)

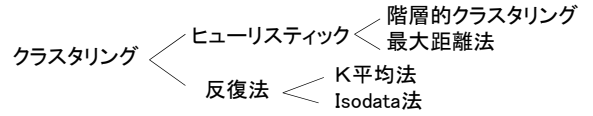


### 原理

領域分割をクラスタリングの手法を用いて解く

### 特徴

1. 各画素が複数の特徴を持つ場合でも適用可能 (カラー画像, 座標値や模様特徴などの利用)
2. 領域の個数が3以上でも適用可能



### クラスタリングの定式化

$N$ 個のサンプルパターン集合  $X = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$  を  $K$ 個の部分集合  $C_1, C_2, \dots, C_K$  に次式の評価関数  $J$  が小さくなるように分割

### ユークリッド距離

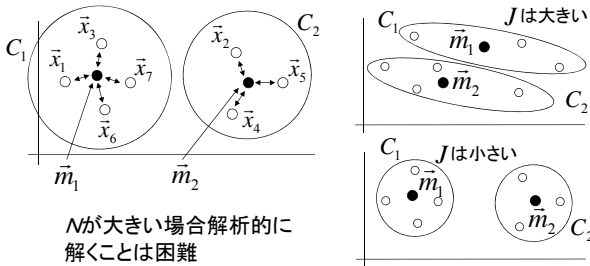
$$J = \sum_{i=1}^K \sum_{\vec{x} \in C_i} \|\vec{x} - \vec{m}_i\|^2 \quad \text{式(18)}$$

クラスタ中心からの距離の合計

### クラスタ中心

$$\vec{m}_i = \frac{1}{N_i} \sum_{\vec{x} \in C_i} \vec{x}$$

$C_i$  に属するデータ個数

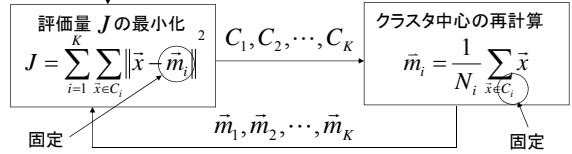


### K平均法

### 原理

クラスタ中心の計算と評価量  $J$  の最小化を交互に行い、反復的に  $J$  を減少させる

初期クラスタ中心  $\vec{m}_1, \vec{m}_2, \dots, \vec{m}_K$



$J$  は次第に減少し極小解に収束する

評価関数の形  $J(C) = J(m(C), C)$   
分割 クラスタ中心

一般的なアルゴリズム ( $k$  は反復回数)

(Step 1) 初期クラスタ中心  $\vec{m}_1(0), \vec{m}_2(0), \dots, \vec{m}_k(0)$  を設定  
 $k = 0$

(Step 2) サンプルパターン集合  $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N$  を次の規則に従ってクラスタ  $C_1(k), C_2(k), \dots, C_k(k)$  に分割

$$\|\vec{x} - \vec{m}_i\| \leq \|\vec{x} - \vec{m}_j\| \text{ for all } j \neq i \Rightarrow \vec{x} \in C_i(k)$$

(Step 3) クラスタ中心を次式で更新する

$$\vec{m}_i(k+1) = \frac{1}{N_i} \sum_{\vec{x} \in C_i(k)} \vec{x}$$

(Step 4) 収束判定を行い、収束していないならStep 2へ戻る

図 17