デジタルメディア処理2

担当: 井尻 敬

デジタルメディア処理２、2017（前期）

4/13	デジタル画像とは	：イントロダクション
4/20	フィルタ処理1	：画素ごとの濃淡変換、線形フィルタ
4/27	フィルタ処理2	：非線形フィルタ，フーリエ変換，ローパスフィルタ，ハイパスフィルタ
5/04	画像の幾何変換１	：アファイン変換
5/11	画像の幾何変換２	：画像の補間，イメージモザイキング
5/18	画像領域分割	：領域拡張法，動的輪郭モデル，グラフカット法
5/25	**前半のまとめ (約30分)と中間試験（約70分）**	
6/01	特徴検出1	：テンプレートマッチング、コーナー検出
6/08	特徴検出2	：DoG特徴量、SIFT特徴量、ハフ変換
6/15	画像認識1	：パターン認識概論，サポートベクタマシン
6/22	画像認識2	：ニューラルネットワーク、深層学習
6/29	画像符号化1	：圧縮率，エントロピー，ランレングス符号化，MH符号化
7/06	画像符号化2	：DCT変換，ウエーブレット変換など
7/13	**後半のまとめ (約30分)と期末試験（約70分）**	

Contents：フィルタ処理２

- 復習：空間フィルタ（線形）
- 空間フィルタ（非線形）
- フーリエ級数展開
- 画像のフーリエ変換
- 周波数フィルタ

復習：空間フィルタ（線形）

空間フィルタ（非線形）

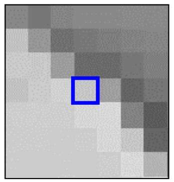
エッジ保存平滑化フィルタ

- 準備：平均と分散
- 実数値の集合$\{x_i | i = 1, \dots, N\}$が与えられたとき、その平均は $\mu = \sum_{i=1}^{N} x_i$, 分散は$\sigma^2 = \sum_{i=1}^{N} (x_i - \mu)^2$で与えられる

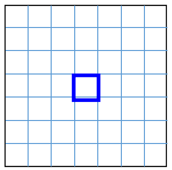
1. 以下の集合の平均と分散を求めよ
   {3,0,3,5,4,3,5,1}

2. 以下の集合AとBどちらが分散が大きい
   A: {3,4,3,4,3,2,2}
   B: {3,5,3,5,3,1,1}

エッジ保存平滑化フィルタ



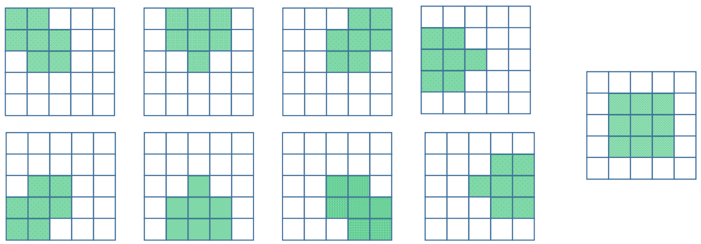
入力画像



出力画像

- 線形平滑化フィルタでは,
  画素(i, j)を計算するため周囲の
  画素の重み付和を計算した

- エッジ保存平滑化フィルタでは、以下9種の領域を考え,
  一番分散の小さな領域の平均値を、その画素の値とする



中央値フィルタ(Median filter)

- 中央値 (median)とは…

数字の集合の代表値
数字の小さい順に並べ、ちょうど中央に位置する値

入力：6, 2, 1, 5, 3, 12, 1000

平均：1/7 x (6+2+1+5+3+12+1000) = 147

中央値： 1, 2, 3, 5, 6, 12, 1000 → 5

中央値と平均値は，用途によって使い分ける
→ 年収など、外れ値の影響が大きい対象には中央値を

中央値フィルタ(Median filter)

- medianFilter.py

TODO 例を置く

＋ 画素(i,j)を中心とする 幅hの窓内の中央値を新しい画素値とする
＋ 外れ値（スパイクノイズ）を除去出来る
＋ 特徴(エッジ)をある程度保存する

バイラテラルフィルタ

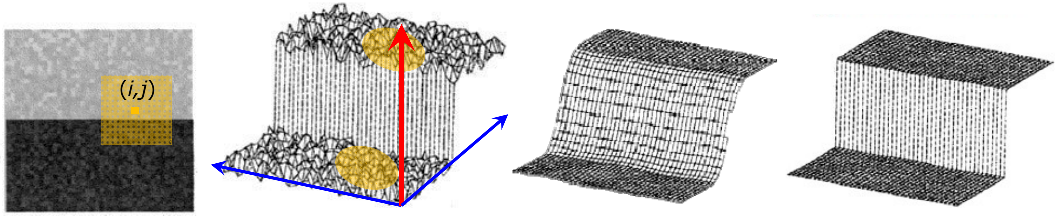
画像中の領域の境界(強いエッジ)をまたがずに平滑化

単純な平滑化　**元画像**　**特徴保存平滑化**

(Gaussian filter)　(bilateral filter)

バイラテラルフィルタ

最も有名な特徴保存フィルタの１つ
空間的距離だけでなく、画素値の差を利用して重み計算



入力画像

Bilateral空間
＋ 位置空間
＋ 値空間

Gaussian filter
位置空間の距離で重み付け
（遠いほど重みを小さく）

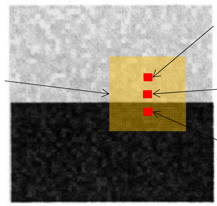
Bilateral filter
Bilateral空間の距離で重み付け
（遠いほど重みを小さく）

画像は [CG-Arts協会 ディジタル画像処理]より

バイラテラルフィルタ

$$I_{new}(\mathbf{x}) = \frac{\sum_{\mathbf{y}\in W_\mathbf{x}} h(\mathbf{x},\mathbf{y})\, I(\mathbf{y})}{\sum_{\mathbf{y}\in W_\mathbf{x}} h(\mathbf{x},\mathbf{y})}$$

$\mathbf{x}$ ：注目画素位置
$\mathbf{y}$ ：局所窓内の画素位置
$W_\mathbf{x}$: $\mathbf{x}$が中心の局所窓



※ 『カーネルh』は窓内の
画素値に依存するので
線形フィルタではない

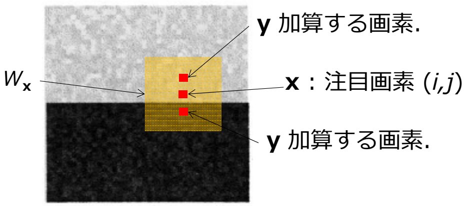
Gaussian filter：$h(\mathbf{x},\mathbf{y}) = G_s(|\mathbf{x}-\mathbf{y}|)$

Bilateral filter：$h(\mathbf{x},\mathbf{y}) = \underbrace{G_s(|\mathbf{x}-\mathbf{y}|)}_{\text{Spatial Kernel}} \cdot \underbrace{G_h(|I(\mathbf{x})-I(\mathbf{y})|)}_{\text{Intensity Kernel}}$

バイラテラルフィルタ

$$I_{new}(\mathbf{x}) = \frac{\sum_{\mathbf{y}\in W_{\mathbf{x}}} h(\mathbf{x},\mathbf{y})\, I(\mathbf{y})}{\sum_{\mathbf{y}\in W_{\mathbf{x}}} h(\mathbf{x},\mathbf{y})}$$

$\mathbf{x}$ ：注目画素位置
$\mathbf{y}$ ：局所窓内の画素位置
$W_{\mathbf{x}}$: $\mathbf{x}$が中心の局所窓



※ 『カーネルh』は窓内の画素値に依存するので線形フィルタではない

Gaussian filter：

$$h(\mathbf{x},\mathbf{y}) = G_s(|\mathbf{x}-\mathbf{y}|)$$

Bilateral filter：

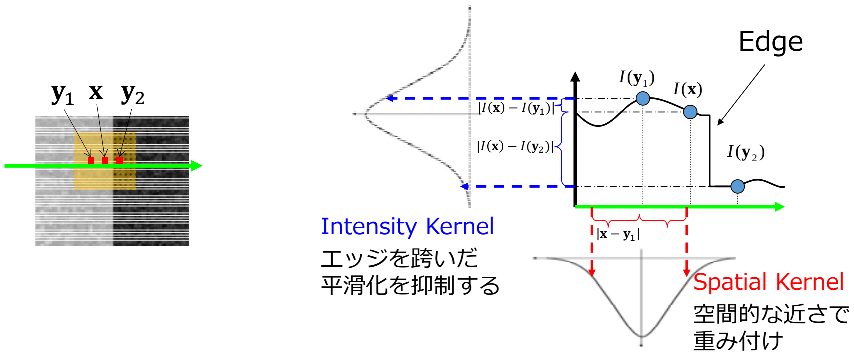
$$h(\mathbf{x},\mathbf{y}) = \underbrace{G_s(|\mathbf{x}-\mathbf{y}|)}_{\text{Spatial Kernel}} \cdot \underbrace{G_h(|I(\mathbf{x})-I(\mathbf{y})|)}_{\text{Intensity Kernel}}$$

G_σは標準偏差σのガウス関数

バイラテラルフィルタ

注目画素位置 $\mathbf{x} = (i,j)$
窓内の画素位置 $\mathbf{y} = (i+m, j+n)$

$$h(\mathbf{x},\mathbf{y}) = G_s(|\mathbf{x}-\mathbf{y}|) \cdot G_h(|I(\mathbf{x})-I(\mathbf{y})|)$$



バイラテラルフィルタ

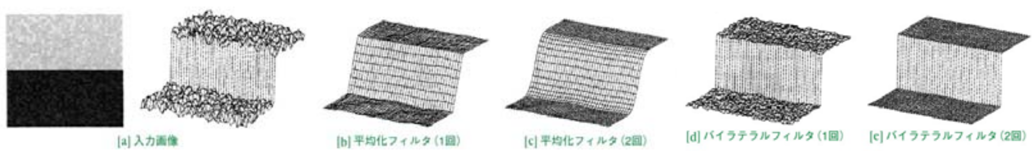
TODO
実装と結果

バイラテラルフィルタ（パラメタ）

$$h(\mathbf{x},\mathbf{y}) = G_s(|\mathbf{x}-\mathbf{y}|) \cdot G_h(|I(\mathbf{x})-I(\mathbf{y})|)$$

パラメータh：平滑化したい領域の輝度値の標準偏差の 0.5-2.0倍程度をよく用いる
複数回適用すると良い結果が出やすい
カラー画像はチャンネル毎でなく、以下を用いて同じ重みを利用するとよい

$$|\mathrm{I}(\mathbf{x}) - \mathrm{I}(\mathbf{y})| = \left\| \begin{pmatrix} R(\mathbf{x}) - R(\mathbf{y}) \\ G(\mathbf{x}) - G(\mathbf{y}) \\ B(\mathbf{x}) - B(\mathbf{y}) \end{pmatrix} \right\|$$



まとめ：空間フィルタ（非線形）

- エッジ保存効果のあるフィルタを紹介した
  - エッジ保存平滑化
  - メディアンフィルタ
  - バイラテラルフィルタ
- 線形フィルタと比べ計算量は大きいが、特殊な効果が得られる

画像は[Shin Yoshizawa撮影]のお台場のガンダムにバイラテラルフィルタを掛けたもの