画像処理

画像間演算，空間フィルタリング

宮崎大学 工学部 情報システム工学科

３年後期
第 ４ 回

本日の内容

画像の加工

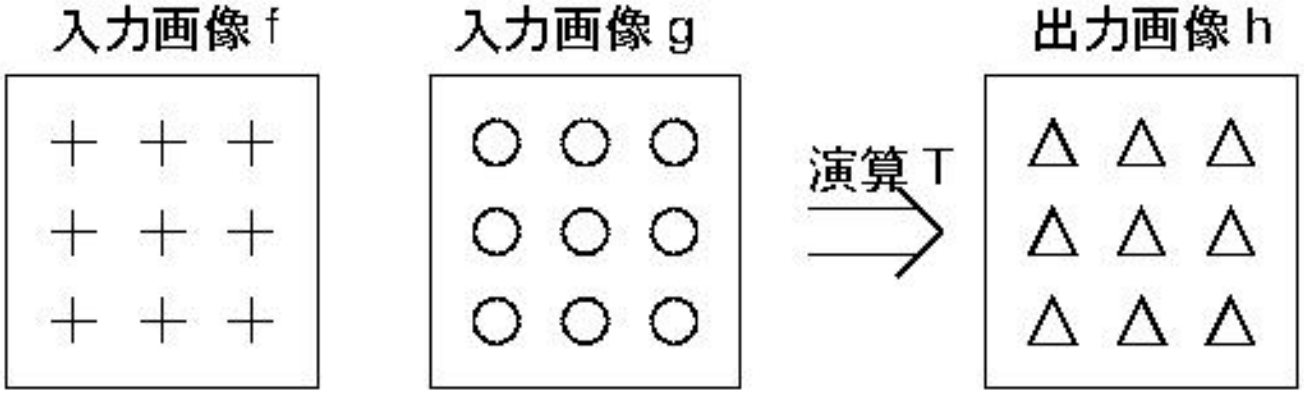
- 画像間演算，クロマキー

空間フィルタリング

- 平滑化
  移動平均，メディアン，エッジ保存平滑化
- エッジ検出
  ソーベル，ラプラシアン
- 鮮鋭化

画像間演算

2つの画像間で，対応する位置にある画素値同士を使って演算を行うこと



$h = T(f, g)$

$\triangle = T(+, \bigcirc)$

演算 T を定義

例：f+g, f-g, f×g, f/g など

画像間の和(f+g)

入力画像 f

入力画像 g

出力画像 h

透過合成したような効果が出る.

画像間の差(f-g)

入力画像 f

入力画像 g

出力画像 h

背景と異なる部分のみを抽出できる.

画像間の差分計算プログラム

```
void img_diff(K_IMAGE *img1, K_IMAGE *img2,
              K_IMAGE *out_img)
{ for(int y = 0; y < k_ysize(out_img); y++){
    for(int x = 0; x < k_xsize(out_img); x++){
      int val =
                ____________________________ ;
      if (val < 0) {
        val = 0;
      } else if (val > 255){
        val = 255;
      }
      (uchar)k_data(out_img)[0][y][x] = val;
  } }
}
```

クロマキー

入力画像 f

入力画像 g

出力画像 h

画像 f の特定の色の部分のみ，画像 g に置き換える．

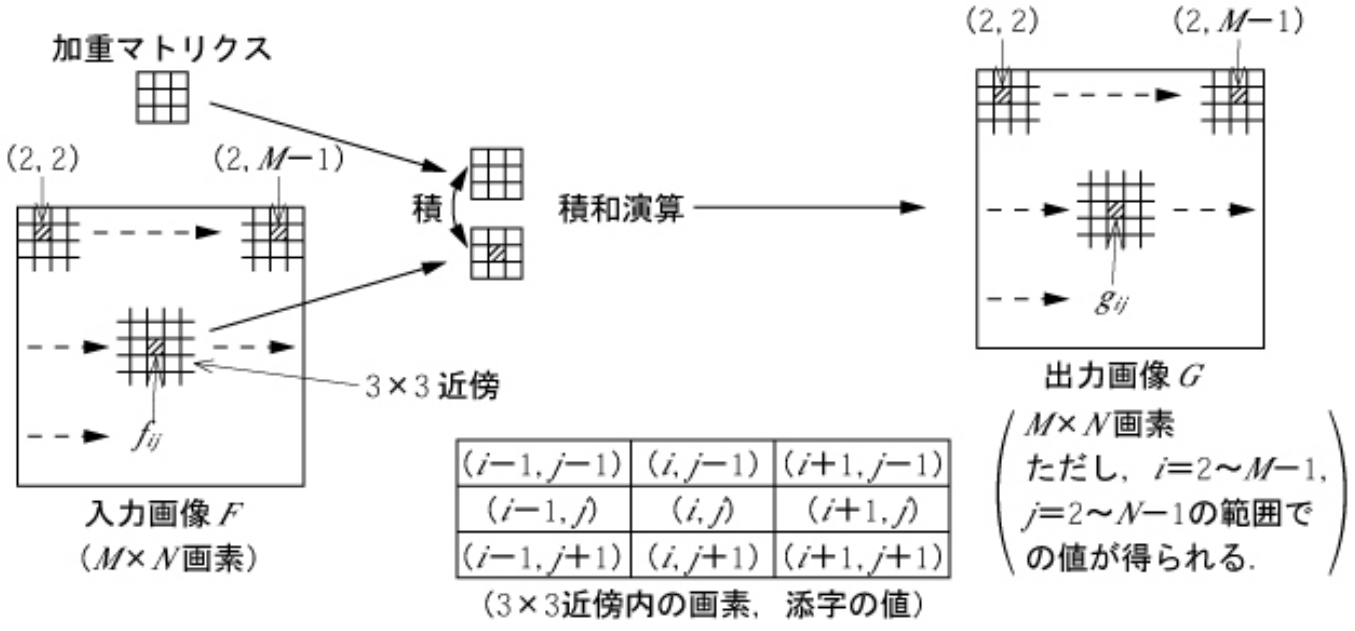
クロマキーのプログラム

img1 の値 255 の画素を img2 で置き換える

```
void chromakey(K_IMAGE *img1, K_IMAGE *img2,
               K_IMAGE *out_img)
{ uchar **iptr1 = (uchar **)k_data(img1)[0];
  uchar **iptr2 = (uchar **)k_data(img2)[0];
  uchar **optr  = (uchar **)k_data(out_img)[0];
  for(int y = 0; y < k_ysize(out_img); y++){
  for(int x = 0; x < k_xsize(out_img); x++){
    if ( iptr1[y][x] == 255 )
         optr[y][x] = ________________________;
    else optr[y][x] = ________________________;
  }}
}
```

空間フィルタリング（局所処理 p.28）

入力画像のある注目画素に対して，その画素と近傍の画素の画素値から出力画像の対応する位置にある画素の画素値を計算する処理



※ フィルタリング：選別されていない元データから必要な情報を取り出す操作

単に「フィルタリング」ともよばれる

平滑化(p.110～)

- 画像上の濃淡変動を滑らかにする処理
- 雑音の低減や画像をぼかす効果を出す

小 ＜＝＝＝＝＝＝＝＝＝＝＝＝＝＞ 大

平滑化度合

移動平均・加重平均 平滑化

- 注目画素の近傍の平均値を出力画素値とする
- 近傍画素との画素値変化がなめらかになる
  ＝平滑化される
- 注目画素との距離に応じて重みづけする手法もある
  ⇒ 加重平均

$\frac{1}{9}\times$

1	1	1
1	1	1
1	1	1

(a)　均一な重み

$\frac{1}{10}\times$

1	1	1
1	2	1
1	1	1

(b)　中央に大きな重み

移動平均と加重平均

原画像

移動平均

加重平均

※単純な移動平均では，フィルタ形状の影響を受ける

移動平均平滑化の例

2	2	1	7	8	8	9
0	2	0	7	8	7	7
2	1	1	9	7	8	8
1	1	2	7	8	0	8
2	0	1	8	7	7	7
1	1	2	7	9	8	7
2	1	0	8	7	9	9

-	-	-	-	-	-	-
-						-
-						-
-						-
-						-
-						-
-	-	-	-	-	-	-

- ３×３の平滑化
- 画像の周囲１画素分は処理しない
- 小数点以下は切捨て

移動平均のプログラム

```
#define B 1
void smooth(K_IMAGE *iimg, K_IMAGE *oimg)
{ uchar **iptr = (uchar **)k_data(iimg)[0];
  uchar **optr = (uchar **)k_data(oimg)[0];
  for(int y=0; y<k_ysize(oimg); y++){
  for(int x=0; x<k_xsize(oimg); x++){
    int sum =0, c = 0;
    for(int j=-B; j<=B; j++){
    for(int i=-B; i<=B; i++){
      if (x+i>=0 && x+i<k_xsize(iimg)
         && y+j>=0 && y+j<k_ysize(iimg))
      {  sum+=iptr[y+j][x+i]; c++; }
    } }
    optr[y][x] = sum/c;
} } }
```

メディアン平滑化(p.113)

- 注目画素の近傍の中央値を出力画素値とする
- 非線形なフィルタ
- 突発的な雑音が含まれていても影響を受けにくい

原画像

移動平均

メディアン

メディアン平滑化の例

2	2	1	7	8	8	9
0	2	0	7	8	7	7
2	1	1	9	7	8	8
1	1	2	7	8	0	8
2	0	1	8	7	7	7
1	1	2	7	9	8	7
2	1	0	8	7	9	9

-	-	-	-	-	-	-
-						-
-						-
-						-
-						-
-						-
-	-	-	-	-	-	-

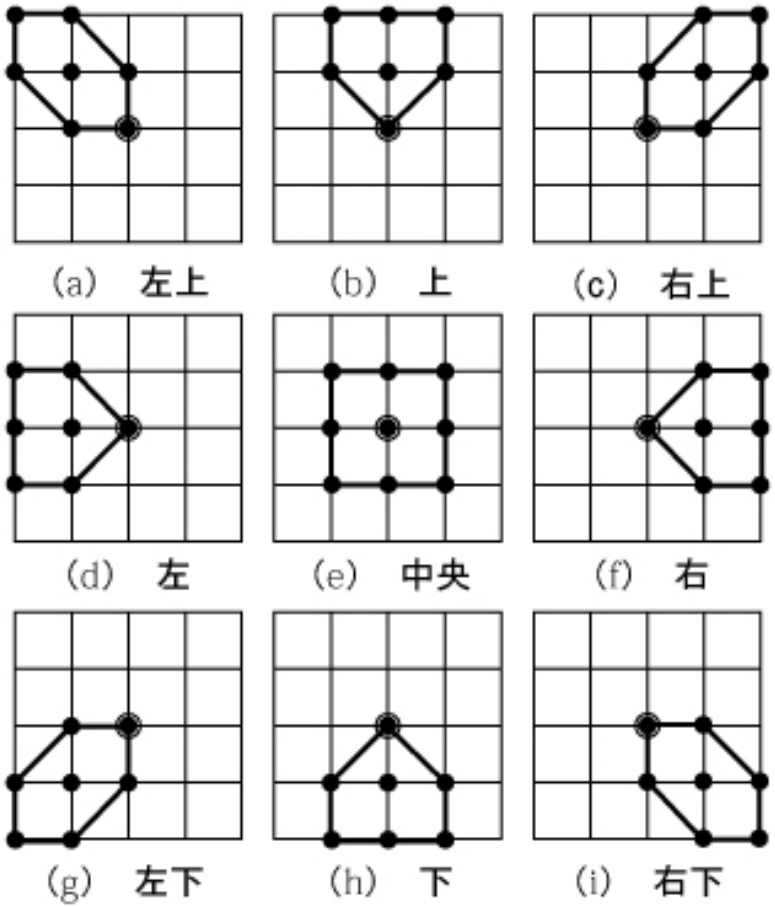
- ３×３の平滑化
- 画像の周囲１画素分は処理しない
- 小数点以下は切捨て

メディアンのプログラム

```
#define B 1
void median(K_IMAGE *iimg, K_IMAGE *oimg)
{ int    pix[(2*B+1)*(2*B+1)];
  for(int y=0; y<k_ysize(iimg); y++){
  for(int x=0; x<k_xsize(iimg); x++) {
      for(j = -B; j <= B; j++){
      for(i = -B, num=0; i <= B; i++){
        if (y+j>=0 && y+j<N && x+i>=0 && x+i<M){
          pix[num] = (uchar)k_data(iimg)[0][y+j][x+i];
          num++;  // フィルタ範囲内のデータをコピー
      } } }
      sort(pix, num); // pix の num 個のデータをソート
      (uchar)k_data(oimg)[0][y][x] = pix[num/2];
                                    // 中央値を代入
} }  }
```

エッジ保存平滑化(p.114)

- 近傍中で画素値の分散が小さい領域の平均を利用
- 画像の大きな濃淡変化を保存したまま平滑化可能



エッジ保存平滑化の結果

原画像

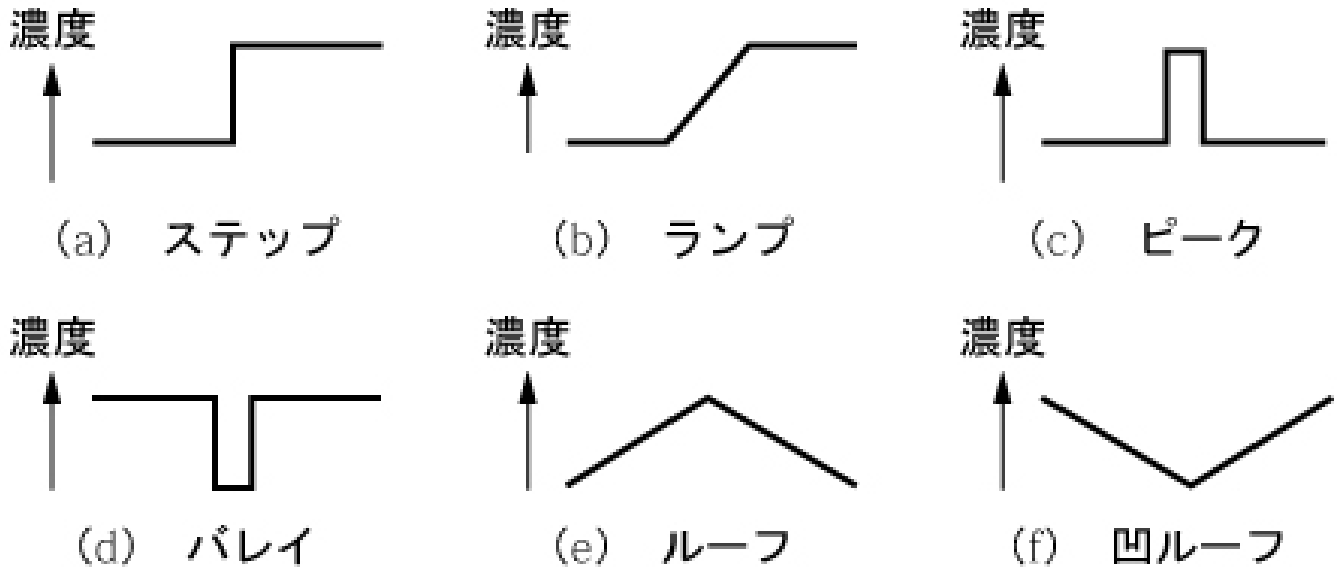
移動平均

エッジ保存

境界がぼけることなく，平滑化されている

エッジ検出 (p.182〜)

エッジ：画像中に現れる大きな明度変化
　対象物の境界／表面の模様／奥行きの変化／影
⇒ 有用な情報を多く含む



ステップエッジ，ランプエッジが検出対象

エッジ検出フィルタ

近傍との濃度値変化が大きい画素に対して大きな画素値を出力する.

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

ソーベル
一次微分

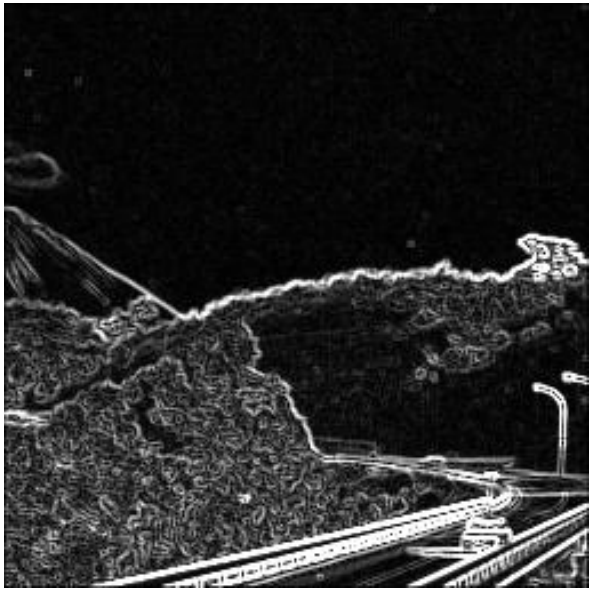
0	1	0
1	−4	1
0	1	0

ラプラシアン
二次微分

※詳細は後の講義で説明

エッジの検出結果

原画像



ソーベル

ラプラシアン

- ソーベルではノイズの影響を受けにくい
- ラプラシアンではエッジ位置が正確

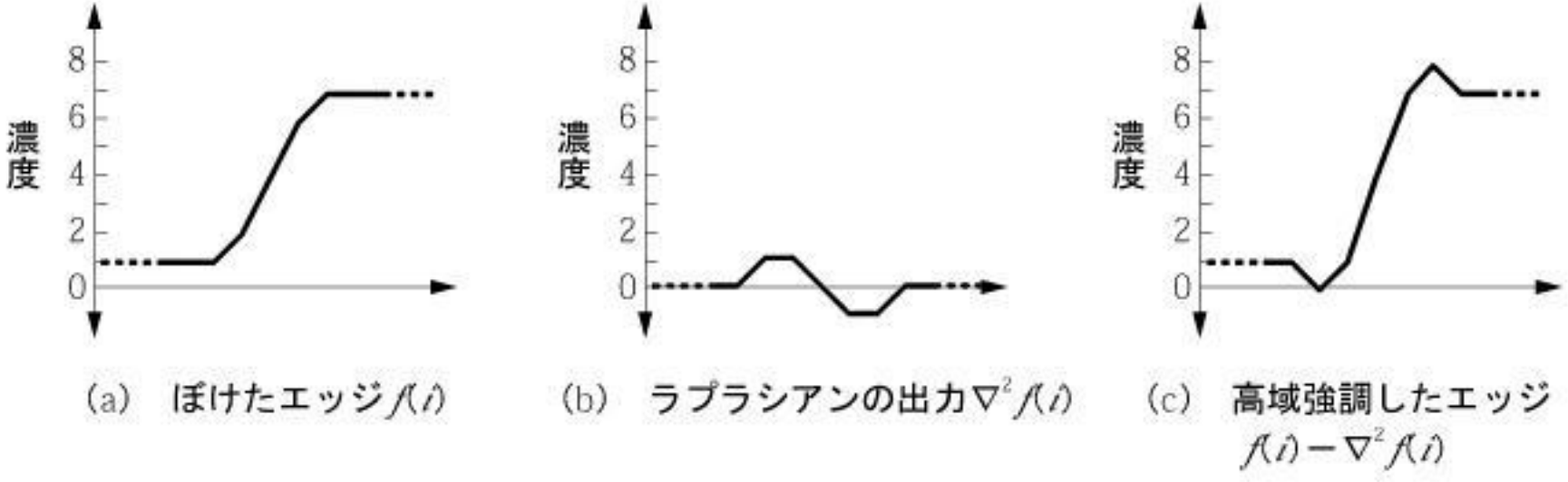
鮮鋭化

ラプラシアンフィルタ

　　エッジの立上がり 正の値

　　エッジの立下がり 負の値

原画像からラプラシアンフィルタの結果を引くことによりエッジの傾きが急峻になる



(a)　ぼけたエッジ $f(i)$　　(b)　ラプラシアンの出力 $\nabla^2 f(i)$　　(c)　高域強調したエッジ $f(i)-\nabla^2 f(i)$

鮮鋭化の結果

- 輪郭部分がくっきりする
- 雑音も同時に強調されてしまう