# 画像処理 課題 2 フィルタリング

21T2166D 渡辺大樹

2023 年 8 月 4 日

## 1　課題の目的・意図

本課題では、ノイズを付加した画像をメジアンフィルタとバイラテラルフィルタの二種類で平滑化し、それぞれのフィルタの特性や処理の得意なノイズを考察、特定することを目的とし行う。

## 2　処理内容

まず目的の画像、今回は図 1 に示す画像にガウシアンノイズとソルト&ペッパーノイズを付加する。これのノイズの除去を、周辺画素の中央値を取るメジアンフィルタと、周辺画素の外れ値を除

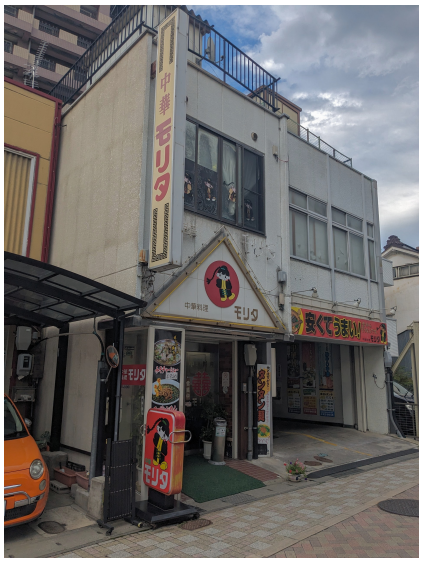


図 1　用意した画像

去し平均化するバイラテラルフィルタの二種類で行っていく。

本演習で用いたメジアンフィルタとバイラテラルフィルタについて初めに少し説明していく。

メジアンフィルタは上記の通り、周辺画素の中央値をとるフィルタになる。平均値ではなく中央

値を取ることで被写体と背景の境界など輝度や色が大きく変わる場所で、出力の値が大きく外れた値になりづらくなる。このため画像のエッジを保存したまま平滑化出来るフィルターとなる。

バイラテラルフィルタは、周辺画素の中で輝度値や色が大きく外れている値を除去するレンジフィルタと、周辺画素の平均値を求める空間フィルタであるガウシアンフィルタの二つをこの順番でかけるフィルタになる。初めにレンジフィルタで外れ値を除去してから平均値を求めることで、こちらも画像のエッジを保存したまま平滑化できるフィルタである。

# 3　処理結果-考察

以下にガウシアンノイズとソルト&ペッパーノイズを付加した図 1 の画像を図 2 に示す。

図 2　ガウシアンノイズ (左図) とソルト&ペッパーノイズ (右図) を付加した図 1

図 2 のようにガウシアンノイズはノイズの輝度値の分布がガウス分布に従ったノイズであり、ソルト&ペッパーノイズは白ないしは黒の極端なノイズになる。

この画像をフィルタで処理した結果を以下に示していく。

## 3.1　ガウシアンノイズ

以下ではガウシアンノイズを付加した画像のフィルタリング結果を示していく。

### 3.1.1　メジアンフィルタ

ガウシアンノイズをメジアンフィルタで処理した結果以下の画像 3 が得られた。右の図が処理後の画像となり、少し小さく見えずらいがかなりノイズが取り除けており、鮮明な画像になっていることが分かる。特に画像中の文字が処理前に比べ視認しやすくなっており、ぷつぷつとしたノイズが取り除けていることが分かる。

図 3　メジアンフィルタでのガウシアンノイズの処理結果

### 3.1.2　バイラテラルフィルタ

次にガウシアンノイズをバイラテラルフィルタで処理した結果以下の画像図 4 が得られた。こち

図 4　バイラテラルフィルタでのガウシアンノイズの処理結果

らも右の図が処理後の画像となる。処理後の画像は背景の空を見ると明確にノイズが取り除けてい

ることが分かるが、前者のメジアンフィルタよりはあまり鮮明なノイズ除去が出来ておらず、正面入口の”モリタ”の文字もあまり復元できていない。

以上より、ガウシアンノイズによる劣化した画像はメジアンフィルタでの平滑化が適していると考えられる。実際に処理内容で考察してみると、白と黒の中間の灰色のノイズが多いガウシアンノイズでは、おそらく外れ値と呼べるようなノイズが出てこず、レンジフィルタでの平滑化がうまく進まなかったのだと考えられる。この点がバイラテラルフィルタの弱点になっていることが分かる。

## 3.2 ソルト&ペッパーノイズ

続いて以下でソルト&ペッパーノイズを付加した画像のフィルタリング結果を示していく。

### 3.2.1 メジアンフィルタ

ソルト&ペッパーノイズをメジアンフィルタで処理した結果以下の画像 5 が得られた。右の図が

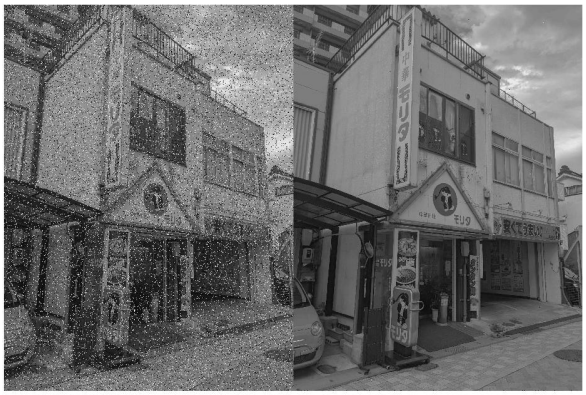

図 5　メジアンフィルタでのソルト&ペッパーノイズの処理結果

処理後の画像となり、少し小さく見えずらいがかなりノイズが取り除けており、鮮明な画像になっていることが分かる。大きく目立つノイズはなくなっており、元の画像の質感を取り戻しつつ、エッジを保存したままの平滑化が出来ている。

### 3.2.2 バイラテラルフィルタ

次にソルト&ペッパーノイズをバイラテラルフィルタで処理した結果以下の画像図 6 が得られた。こちらも右の図が処理後の画像となる。この図では処理前処理後の画像を比較しても分かるような変化がなく、全くノイズが取り除けていないように見える。実際にノイズが取り除けているの

図 6　バイラテラルフィルタでのソルト&ペッパーノイズの処理結果

かどうかこの 2 画像の差を取った画像を図 7 に示す。

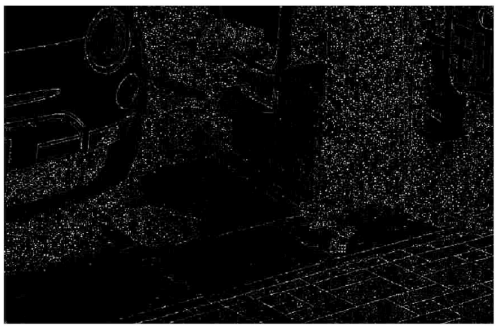

図 7　図 6 の左右の画像差分

可視性を上げるため、ある程度差を見やすくし、図 1 の左下にある車部分を大きく拡大している。これを見ると確かに処理が施されておりノイズが除去されていることが分かるが、ただ肉眼で見ると全く変化がないように見え、平滑化ができたとは言えない結果になってしまった。

以上の結果よりソルト&ペッパーノイズにより劣化した画像の平滑化もメジアンフィルタが適していることが考えられる。

私の演習前の予想としては白黒のノイズが外れ値となり、バイラテラルフィルタをかけたときのレンジフィルタにより除外されると思っていたがあまりうまく動かなかった。演習に用いたコードに誤りがあったり、適切ではない窓半径を用いてしまったことも考えられるが、私の力ではここまでの結果しか得られなかった。

最後にこの実験にて用いたコードと条件を示す。ガウシアンフィルタの標準偏差は 1、メジアンフィルタの窓サイズは 1 で行った。

ソースコード 1　filtering.m

```
I = im2double( imread('./img/Morita.jpg') );
Iycc = rgb2ycbcr( I );
Y = Iycc(:,:,1);

rate_sp = 0.1;
Y_rate_sp = imnoise( Y, 'salt & pepper', rate_sp );

figure(1), imshow(Y_rate_sp);

std_gauss = 0.1;
Y_gauss = imnoise( Y, 'gaussian', 0, std_gauss^2 );

figure(2), imshow(Y_gauss);

m_r = 1; %小さいとノイズを取り除けず､大きいとエッジがなくなる

% 空間フィルタ(ガウシアンフィルタを使用)
ss = 1; % 空間フィルタの標準偏差 (standard deviation for spatial)
g_r = round(3*ss);

Ks = fspecial('gaussian', 2*[g_r,g_r]+1, ss);
Ks = Ks / sum( Ks(:) );

% レンジフィルタ　関数の定義(ガウス関数を使用)
sr = 0.1; % レンジフィルタの標準偏差 (standard deviation for range)
k_r = @(X,xi,sr) exp( (-0.5/sr^2)*(X-xi).^2 );


T = 10;
psi = @(X,t,T) max(0,-abs(X-t)/(1/T)+1);

% --- solt & pepper noise -> bilateral filter ---

Nume = zeros( size(Y_rate_sp) );

for t = 0:1/T:1
    PI = psi( Y_rate_sp, t, T ) .* Y_rate_sp;
    PI_s = imfilter( PI, Ks, 'corr', 'replicate' );
    Nume = Nume + k_r( Y_rate_sp, t, sr ) .* PI_s;
end

```

```
Deno = zeros( size(Y_rate_sp) );
U = ones( size(Y_rate_sp) );

for t = 0:1/T:1
    PU = psi( Y_rate_sp, t, T ) .* U;
    PU_s = imfilter( PU, Ks, 'corr', 'replicate' );
    Deno = Deno + k_r( Y_rate_sp, t, sr ) .* PU_s;
end

Y_sp_bil = Nume ./ Deno;
figure(3), imshow( [Y_rate_sp, Y_sp_bil] );
figure(13), imshow( Y_sp_bil-Y_rate_sp );
K = im

% --- gauss noise -> bilateral filter ---

Nume = zeros( size(Y_gauss) );

for t = 0:1/T:1
    PI = psi( Y_gauss, t, T ) .* Y_gauss;
    PI_s = imfilter( PI, Ks, 'corr', 'replicate' );
    Nume = Nume + k_r( Y_gauss, t, sr ) .* PI_s;
end

Deno = zeros( size(Y_gauss) );
U = ones( size(Y_gauss) );

for t = 0:1/T:1
    PU = psi( Y_gauss, t, T ) .* U;
    PU_s = imfilter( PU, Ks, 'corr', 'replicate' );
    Deno = Deno + k_r( Y_gauss, t, sr ) .* PU_s;
end

Y_gauss_bil = Nume ./ Deno;

figure(4), imshow( [Y_gauss, Y_gauss_bil] );

Y_sp_med = medfilt2( Y_rate_sp, [2*m_r+1, 2*m_r+1] );
Y_gauss_med = medfilt2( Y_gauss, [2*m_r+1, 2*m_r+1]);

figure(5), imshow( [Y_rate_sp, Y_sp_med] );
figure(6), imshow( [Y_gauss, Y_gauss_med]);
```