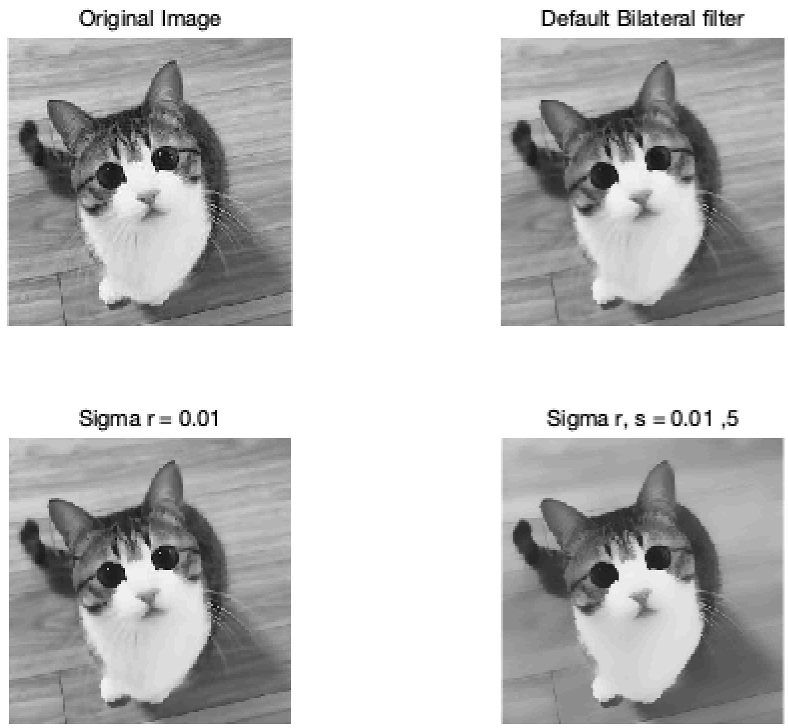
Table of Contents

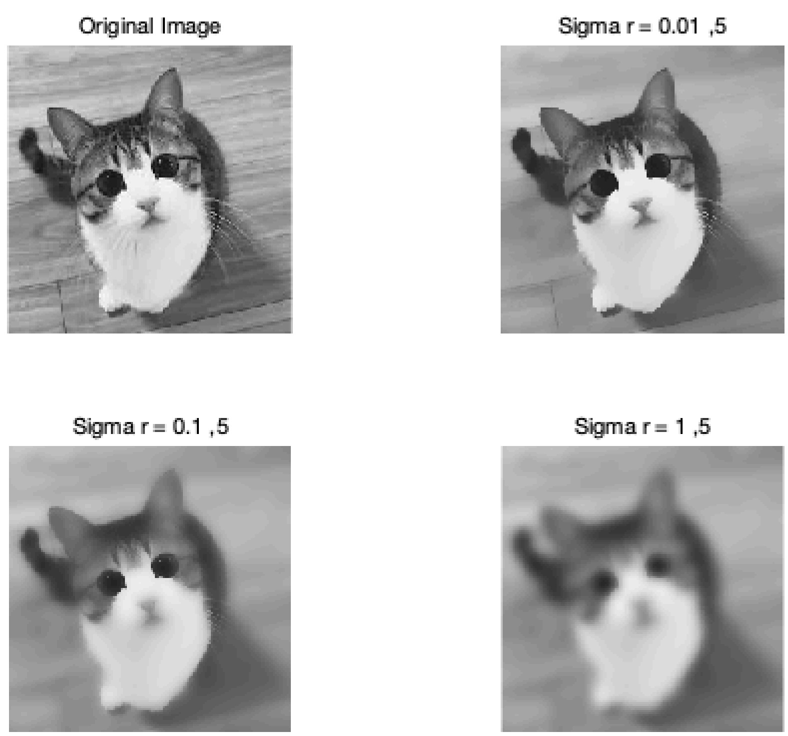
Using matlab imbilatfilt function 1
Using bilateral on given image 4
Iteration Bilateral 5
Do histeq and percentage ahead of Bilateral 8
Writing bilateral by myself 9

Using matlab imbilatfilt function

```
clc;
close all;
clear;

g= im2double(imread('hw_img3.jpeg'));
g = rgb2gray(g);

%imbilatfilt(g)
G1 = imbilatfilt(g);

%imbilatfilt(g, sigma_r), the 2nd arg is for the gray value variance
sigma_r = 0.01;
G2 = imbilatfilt(g,sigma_r);

%imbilatfilt(g, sigma_r, sigma_s), the 3rd arg is for spatial variance
sigma_s = 5;
G3 = imbilatfilt(g,sigma_r,sigma_s);

figure()
subplot(2,2,1)
axis on;
imshow(g)
title('Original Image')

subplot(2,2,2)
axis on;
imshow(G1)
title('Default Bilateral filter')

subplot(2,2,3)
axis on;
imshow(G2)
title(['Sigma r = ', num2str(sigma_r)])

subplot(2,2,4)
axis on;
imshow(G3)
title(['Sigma r, s = ', num2str(sigma_r), ' ,', num2str(sigma_s)]);
% We can see from the results that after
% It's because sigma_r is very small, no smoothing occurs on the edge.
% As a consequence, increasing the spatial sigma has no
```

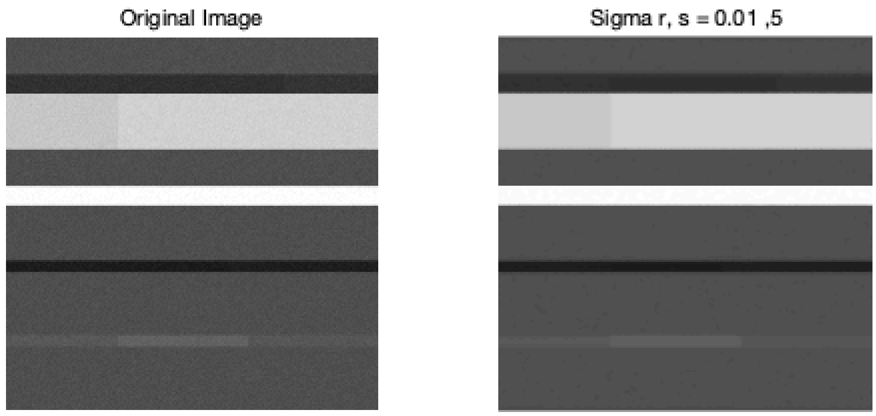
```
% consequence on a edge as long as the range sigma is less than
% its amplitude.

% here we see the effection of sigma_r
figure()

r1 = 0.01;
r2 = 0.1;
r3 = 1;

subplot(2,2,1)
axis on;
imshow(g)
title('Original Image')

subplot(2,2,2)
axis on;
imshow(imbilatfilt(g, r1, sigma_s))
title(['Sigma r = ', num2str(r1), ' ,', num2str(sigma_s)])

subplot(2,2,3)
axis on;
imshow(imbilatfilt(g, r2, sigma_s))
title(['Sigma r = ', num2str(r2), ' ,', num2str(sigma_s)])

subplot(2,2,4)
axis on;
imshow(imbilatfilt(g, r3, sigma_s))
title(['Sigma r = ', num2str(r3), ' ,', num2str(sigma_s)]);

% Draw a conclusion, 1) when the sigma_r increases, the guassian
 function
% will be fattlen, which means the pixels which have large gray
% variance will also be considered. Therefore, in order to
% maintain the edge information of the image, it' would be better to
 use
% a small sigma_r; 2) the sigma_s represeat the variance on spatial,
% the larger sigma_s, the more flatten guassian mask, the more
 blurring.
```


Original Image
Default Bilateral filter
Sigma r = 0.01
Sigma r, s = 0.01 ,5


Original Image
Sigma r = 0.01 ,5
Sigma r = 0.1 ,5
Sigma r = 1 ,5

Using bilateral on given image

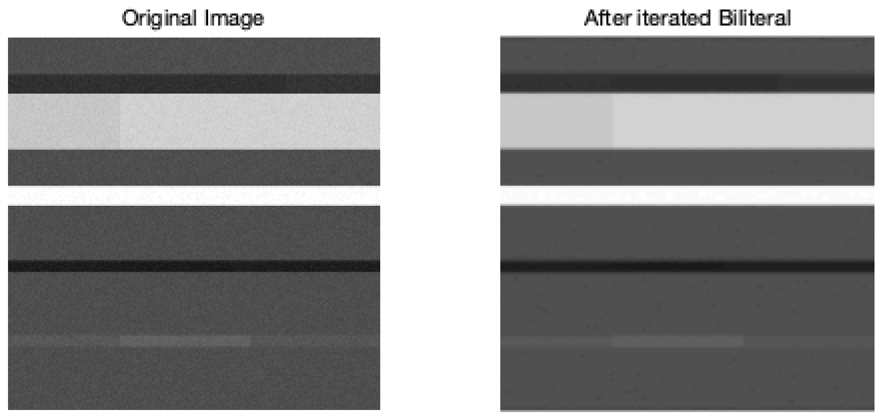
```
clear;
close all;
clc;

cells = load("cell_profile_n.mat");
cells_r = uint8(cells.cell_profile_n);
input = double(cells_r)/255.0;
sigma_s=5;
sigma_r=0.01;

output = imbilatfilt(input,sigma_r,sigma_s);

subplot(1,2,1)
axis on;
imshow(input)
title('Original Image')

subplot(1,2,2)
axis on;
imshow(output)
title(['Sigma r, s = ', num2str(sigma_r), ' ,', num2str(sigma_s)]);

% we can see that if we apply the bilateral filter directly on the
% cell_noise image, it just blurs the image and the effection
% is not so obvious.

% Therefore, in order to show more detial of the image and
% demonstrate the performance of bilateral filter, the following thing
% will be done in the next part:
%1) a local histeq / percentage is applied after the bilateral to
% enhance the detail of input.
%2)a local histeq / percentage is applied ahead of the bilateral
%to enhance the detail of input.
```

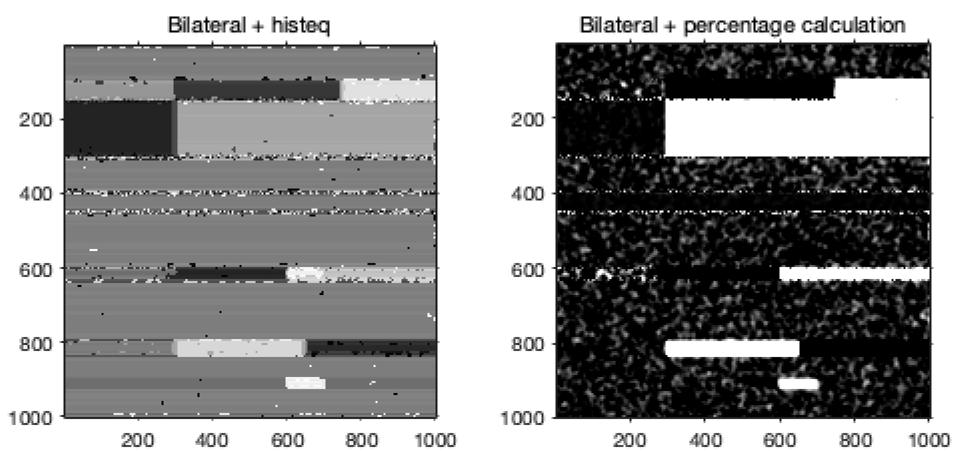


Iteration Bilateral

```
clear;
close all;
clc;

cells = load("cell_profile_n.mat");
cells_r = uint8(cells.cell_profile_n);
input = double(cells_r)/255.0;
sigma_s=5;

sigma_r= 0.1;
for i = 1:3
    output = imbilatfilt(input,sigma_r,sigma_s);
end

figure();
subplot(1,2,1)
axis on
imshow(input);
title('Original Image')
subplot(1,2,2)
axis on
imshow(output);
title('After iterated Biliteral')
```

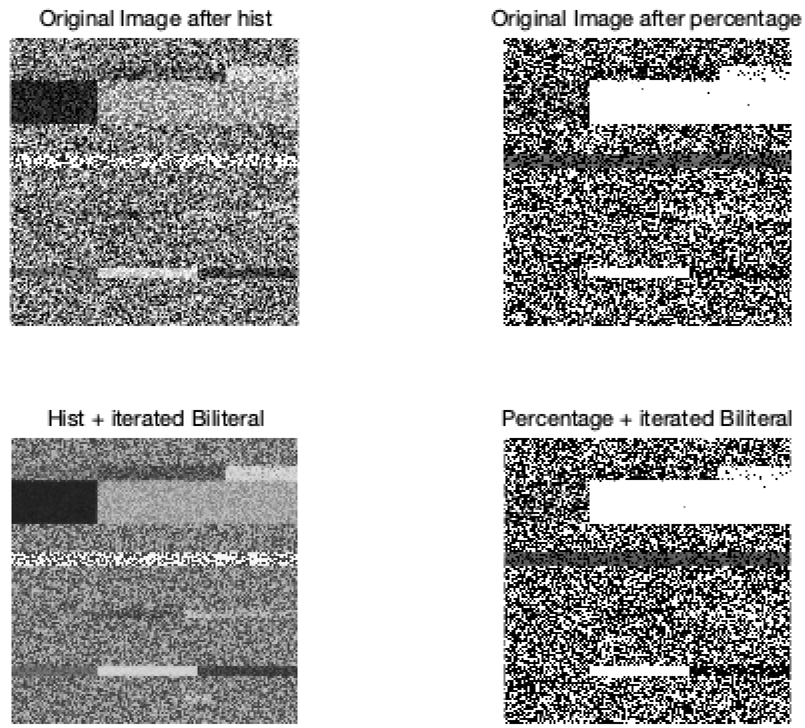
```
% we do histeq or calculate percentage

% local histeq
for i = 1:size(output,1)
    output_hist(i,:) = histeq(output(i,:));
end

% percentage
steady_state = mean(output(:,1:299), 2);
percentage_change = 100 * (output - steady_state) ./ steady_state;

figure()
subplot(1,2,1)
imshow(output_hist)
axis on
title('Bilateral + histeq')
subplot(1,2,2)
imshow(percentage_change)
axis on
title('Bilateral + percentage calculation')

% In this part we try multiple iteration of the bilateral. As the
% iteration number increaseing, the some edge of image will be
% further blured, and results in a "cartoon - like" image.

% As illustrated by the results, 1)the histeq result can show more
 detial
% and have a even background, however it has artificially blurred
 edge,
% which is because the bilateral filter changed the distribution
% of the gray value of pixels. 2) the percentage result does't have
% "spike" noise. However, it losses some detail and have a uneven
% background, which is because the bilateral change the value of
% original background and introduce "varience" on pixel values
```


Original Image
After iterated Biliteral


Bilateral + histeq
Bilateral + percentage calculation
200
400
600
800
1000

Do histeq and percentage ahead of Bilateral

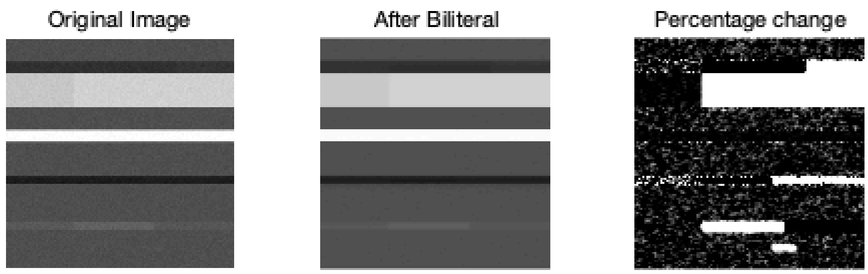
```
clear;
close all;
clc;

cells = load("cell_profile_n.mat");
input1 = double(cells.cell_profile_n);
input1 = double(input1)/255.0;
input2 = input1;

for i = 1:size(input1,1)
    input1(i,:) = histeq(input1(i,:));
end

steady_state = mean(input2(:,1:299), 2);
percentage_change = 100 * (input2 - steady_state) ./ steady_state;
input2 = percentage_change;

sigma_s=5;
sigma_r=0.1;

for i = 1:3
    output1 = imbilatfilt(input1,sigma_r,sigma_s);
    output2 = imbilatfilt(input2,sigma_r,sigma_s);
end

figure();
subplot(2,2,1)
axis on
imshow(input1);
title('Original Image after hist')
subplot(2,2,2)
axis on
imshow(input2);
title('Original Image after percentage')
subplot(2,2,3)
axis on
%imshow(mat2gray(output(Wsize+1:r+Wsize,Wsize+1:c+Wsize)));
imshow(output1);
title('Hist + iterated Biliteral')
subplot(2,2,4)
axis on
%imshow(mat2gray(output(Wsize+1:r+Wsize,Wsize+1:c+Wsize)));
imshow(output2);
title('Percentage + iterated Biliteral')

% Obviously, if we using enhancement ahead of time, the noise will be
% enhanced and change it's form -> no longer be a guassian noise,
% therefore, these enhancement should be added after the filter
```



Writing bilateral by myself

```
clear;
close all;
clc;

cells = load("cell_profile_n.mat");
cells_r = uint8(cells.cell_profile_n);
input = double(cells_r)/255.0;
Wsize=10;
% variance of the image
sigma_s=5;
sigma_r=0.1;
% using bilateral filter
output=Mybilateral(input,sigma_r,sigma_s,Wsize);

steady_state = mean(output(:,1:299), 2);
percentage_change = 100 * (output - steady_state) ./ steady_state;

figure();
subplot(1,3,1)
axis on
imshow(input);
title('Original Image')
subplot(1,3,2)
```

```
axis on
imshow(output);
title('After Biliteral')
subplot(1,3,3)
axis on
imshow(percentage_change);
title('Percentage change')

% Here we implement our own bilateral filter function. The detail
% comments are on the Mybilateral.m
```



Published with MATLAB® R2020a