Contents

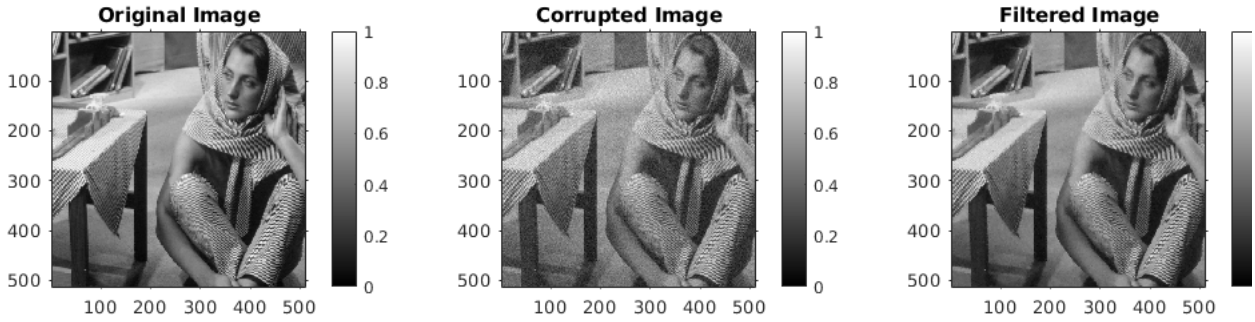
MyMainScript

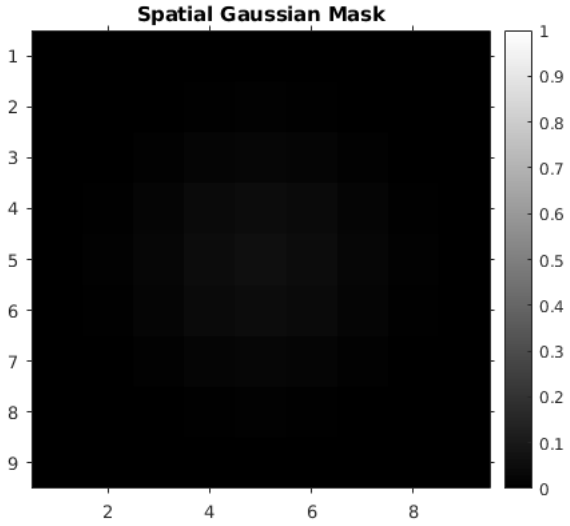
```
tic;
```

Barbara

```
imageData = load('../data/barbara.mat');
image = imageData.imageOrig;
[len, wid] = size(image);
corruptedImage = image + 0.05*(max(max(image))-min(min(image)))*randn(len);

sigmaSpace = 1.6;
sigmaIntensity = 13.5;
windowSize = 4;
out = myBilateralFiltering(corruptedImage, sigmaSpace, sigmaIntensity, windowSize);
rmsd = myRmsd(out,image);
display(rmsd);
myDisplayThreeImage(image,corruptedImage,out);

out1 = myBilateralFiltering(corruptedImage, 0.9*sigmaSpace, sigmaIntensity, windowSize);
rmsd1 = myRmsd(out1,image);
display(rmsd1);

out2 = myBilateralFiltering(corruptedImage, 1.1*sigmaSpace, sigmaIntensity, windowSize);
rmsd2 = myRmsd(out2,image);
display(rmsd2);

out3 = myBilateralFiltering(corruptedImage, sigmaSpace, 0.9*sigmaIntensity, windowSize);
rmsd3 = myRmsd(out3,image);
display(rmsd3);

out4 = myBilateralFiltering(corruptedImage, sigmaSpace, 1.1*sigmaIntensity, windowSize);
rmsd4 = myRmsd(out4,image);
display(rmsd4);

figure;
filter = fspecial('gaussian',2*windowSize+1 ,sigmaSpace);
imshow(filter,'InitialMagnification','fit');
title("Spatial Gaussian Mask")
colormap(gray(200));
axis on;
colorbar;
```

```
Optimal -> rmsd = 3.2807 (sigmaSpace = 1.6, sigmaIntensity = 13.5, windowSize = 9)


0.9*sigmaSpace -> rmsd1 = 3.2824 (sigmaSpace = 1.6*0.9, sigmaIntensity = 13.5, windowSize = 9)


1.1*sigmaSpace -> rmsd2 = 3.2847 (sigmaSpace = 1.6*0.1, sigmaIntensity = 13.5, windowSize = 9)


0.9*sigmaIntensity -> rmsd3 = 3.3102 (sigmaSpace = 1.6, sigmaIntensity = 13.5*0.9, windowSize = 9)


1.1*sigmaIntensity -> rmsd4 = 3.2895 (sigmaSpace = 1.6, sigmaIntensity = 13.5*1.1, windowSize = 9)
```



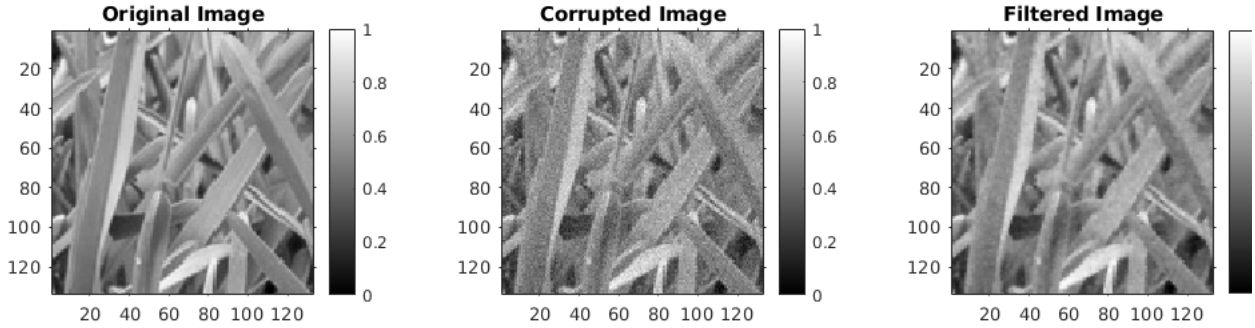


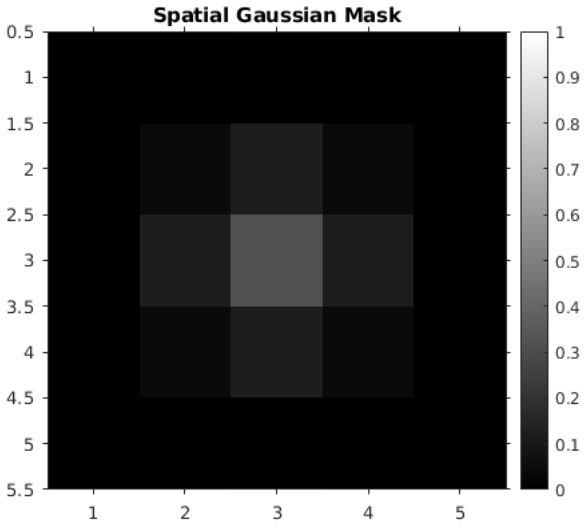
Grass

```
image = im2double(imread('../data/grass.png'));
[len, wid] = size(image);
corruptedImage = image + 0.05*(max(max(image))-min(min(image)))*randn(len);

sigmaSpace = 0.71;
sigmaIntensity = 0.31;
windowSize = 2;
out = myBilateralFiltering(corruptedImage, sigmaSpace, sigmaIntensity, windowSize);
rmsd = myRmsd(out,image);
display(rmsd);
myDisplayThreeImage(image,corruptedImage,out);

out1 = myBilateralFiltering(corruptedImage, 0.9*sigmaSpace, sigmaIntensity, windowSize);
rmsd1 = myRmsd(out1,image);
display(rmsd1);

out2 = myBilateralFiltering(corruptedImage, 1.1*sigmaSpace, sigmaIntensity, windowSize);
rmsd2 = myRmsd(out2,image);
display(rmsd2);

out3 = myBilateralFiltering(corruptedImage, sigmaSpace, 0.9*sigmaIntensity, windowSize);
rmsd3 = myRmsd(out3,image);
display(rmsd3);

out4 = myBilateralFiltering(corruptedImage, sigmaSpace, 1.1*sigmaIntensity, windowSize);
rmsd4 = myRmsd(out4,image);
display(rmsd4);

figure;
```

```
filter = fspecial('gaussian',2*windowSize+1 ,sigmaSpace);
imshow(filter,'InitialMagnification','fit');
title("Spatial Gaussian Mask")
colormap(gray(200));
axis on;
colorbar;
```

```
Optimal -> rmsd = 0.0289 (sigmaSpace = 0.71, sigmaIntensity = 0.31, windowSize = 5)

0.9*sigmaSpace -> rmsd1 = 0.0293 (sigmaSpace = 0.71*0.9, sigmaIntensity = 0.31, windowSize = 5)

1.1*sigmaSpace -> rmsd2 = 0.0290 (sigmaSpace = 0.71*0.1, sigmaIntensity = 0.31, windowSize = 5)

0.9*sigmaIntensity -> rmsd3 = 0.0289 (sigmaSpace = 0.71, sigmaIntensity = 0.31*0.9, windowSize = 5)

1.1*sigmaIntensity -> rmsd4 = 0.0289 (sigmaSpace = 0.71, sigmaIntensity = 0.31*1.1, windowSize = 5)
```



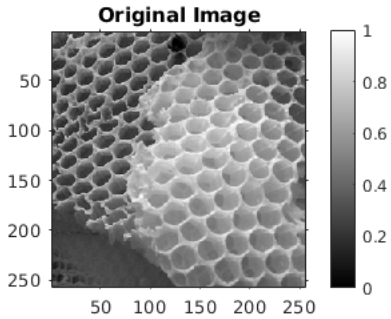


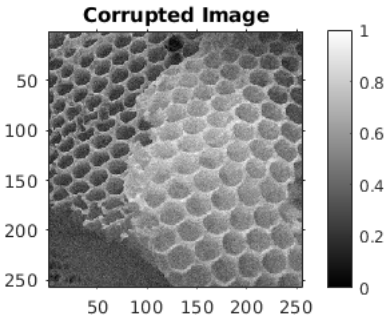
Honey Comb Real

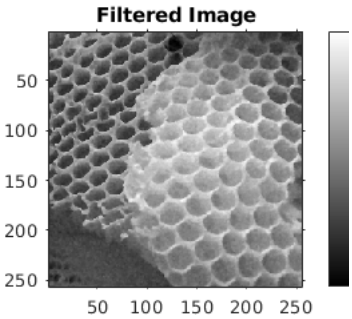
```
image = im2double(imread('../data/honeyCombReal.png'));
[len, wid] = size(image);
corruptedImage = image + 0.05*(max(max(image))-min(min(image)))*randn(len);
```

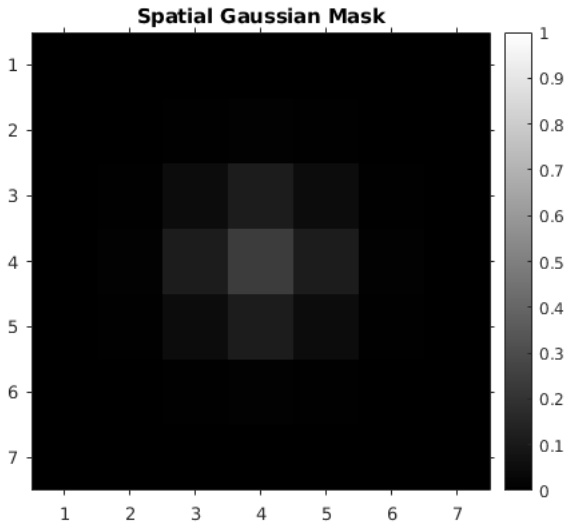
```
sigmaSpace = 0.82;
sigmaIntensity = 0.27;
windowSize = 3;
out = myBilateralFiltering(corruptedImage, sigmaSpace, sigmaIntensity, windowSize);
rmsd = myRmsd(out,image);
display(rmsd);
myDisplayThreeImage(image,corruptedImage,out);

out1 = myBilateralFiltering(corruptedImage, 0.9*sigmaSpace, sigmaIntensity, windowSize);
rmsd1 = myRmsd(out1,image);
display(rmsd1);

out2 = myBilateralFiltering(corruptedImage, 1.1*sigmaSpace, sigmaIntensity, windowSize);
rmsd2 = myRmsd(out2,image);
display(rmsd2);

out3 = myBilateralFiltering(corruptedImage, sigmaSpace, 0.9*sigmaIntensity, windowSize);
rmsd3 = myRmsd(out3,image);
display(rmsd3);

out4 = myBilateralFiltering(corruptedImage, sigmaSpace, 1.1*sigmaIntensity, windowSize);
rmsd4 = myRmsd(out4,image);
display(rmsd4);

figure;
filter = fspecial('gaussian',2*windowSize+1 ,sigmaSpace);
imshow(filter,'InitialMagnification','fit');
title("Spatial Gaussian Mask")
colormap(gray(200));
axis on;
colorbar;

toc;
```

```
Optimal -> rmsd = 0.0288 (sigmaSpace = 0.82, sigmaIntensity = 0.27, windowSize = 7)


0.9*sigmaSpace -> rmsd1 = 0.0292 (sigmaSpace = 0.82*0.9, sigmaIntensity = 0.27, windowSize = 7)


1.1*sigmaSpace -> rmsd2 = 0.0289 (sigmaSpace = 0.82*0.1, sigmaIntensity = 0.27, windowSize = 7)


0.9*sigmaIntensity -> rmsd3 = 0.0288 (sigmaSpace = 0.82, sigmaIntensity = 0.27*0.9, windowSize = 7)


1.1*sigmaIntensity -> rmsd4 = 0.0290 (sigmaSpace = 0.82, sigmaIntensity = 0.27*1.1, windowSize = 7)

Elapsed time is 12.311102 seconds.
```









Published with MATLAB© R2018a