Table of Contents

MyMainScript

```
tic;
% Your code here
```

Grass

The optimal value of sigma obtained is 1.6 and corresponding rmsd is 9.1626

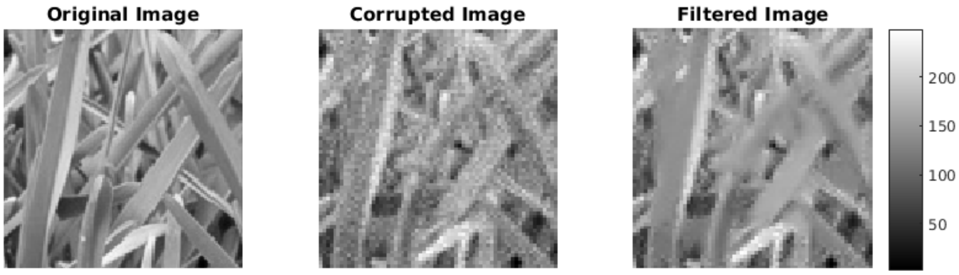
- Value of RMSD for 0.9sigma = 9.2467

- Value of RMSD for 1.1sigma = 9.2914

```
image = imread('../data/grass.png');

[final_img, noisy_img, downsampled_img] = getFilteredImage(image,
 1.6);
% Uncomment this line to get optimal RMSD
% [hf, min_rms] = getOptimalRMSD(downsampled_img, noisy_img,
 1.5:0.1:2);


figure('Renderer', 'painters', 'Position',[10 10 900 600]);
subplot(1,3,1);
imshow(uint8(image), 'DisplayRange', []);
title('Original Image');
subplot(1, 3, 2);
imshow(uint8(noisy_img), 'DisplayRange', []);
title('Corrupted Image');
x = subplot(1, 3, 3);
pos = get(x, 'position');
imshow(uint8(final_img), 'DisplayRange', []);
title('Filtered Image');
colorbar;
set(x, 'position', pos);
```



Honey Comb

The optimal value of sigma obtained is 1.7 and corresponding rmsd is 9.3342

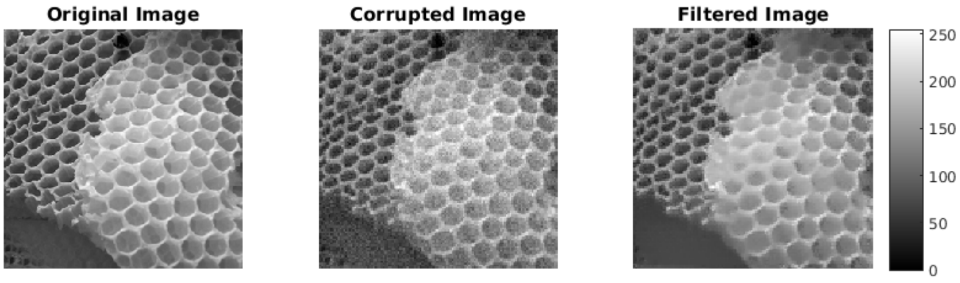
- Value of RMSD for 0.9sigma = 9.6119
- Value of RMSD for 1.1sigma = 9.5690

```
image = imread('../data/honeyCombReal.png');

[final_img, noisy_img, downsampled_img] = getFilteredImage(image,
 1.7);
% [hf, min_rms] = getOptimalRMSD(downsampled_img, noisy_img,
 1.5:0.1:2);


figure('Renderer', 'painters', 'Position',[10 10 900 600]);
subplot(1,3,1);
imshow(uint8(image), 'DisplayRange', []);
title('Original Image');
subplot(1, 3, 2);
imshow(uint8(noisy_img), 'DisplayRange', []);
title('Corrupted Image');
x = subplot(1, 3, 3);
pos = get(x, 'position');
imshow(uint8(final_img), 'DisplayRange', []);
title('Filtered Image');
colorbar;
```

```
set(x, 'position', pos);
```



Barbara

The optimal value of sigma obtained is 0.7 and corresponding rmsd is 2.83

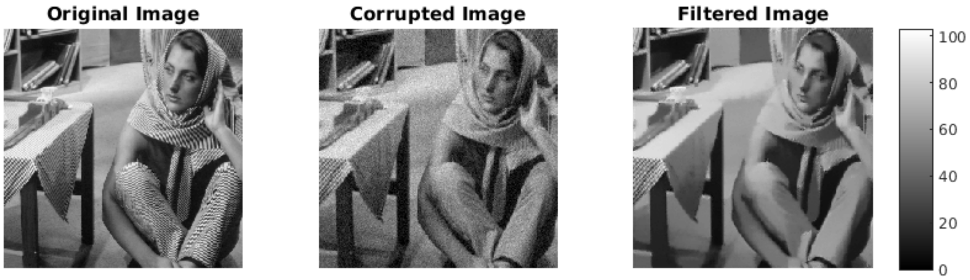
- Value of RMSD for 0.9sigma = 2.8676
- Value of RMSD for 1.1sigma = 2.8913

```
image = load('../data/barbara.mat');
image = image.imageOrig;

[final_img, noisy_img, downsampled_img] = getFilteredImage(image,
 0.7);
% [hf, min_rms] = getOptimalRMSD(downsampled_img, noisy_img,
 0.5:0.1:1);


figure('Renderer', 'painters', 'Position',[10 10 900 600]);
subplot(1,3,1);
imshow(uint8(image), 'DisplayRange', []);
title('Original Image');
subplot(1, 3, 2);
imshow(uint8(noisy_img), 'DisplayRange', []);
title('Corrupted Image');
x = subplot(1, 3, 3);
pos = get(x, 'position');
```

```
imshow(uint8(final_img), 'DisplayRange', []);
title('Filtered Image');
colorbar;
set(x, 'position', pos);
```



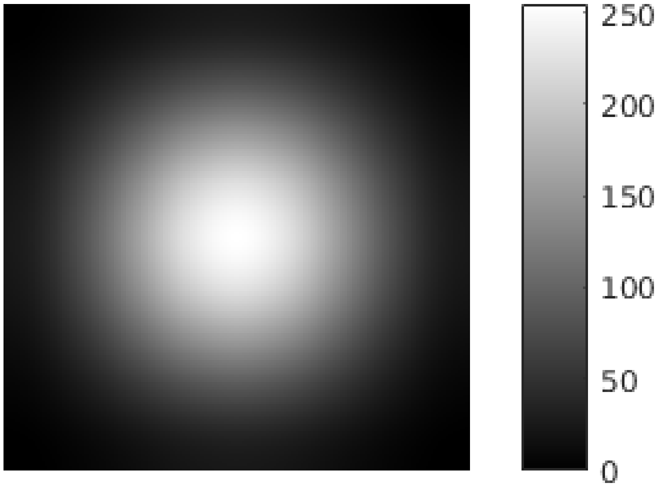
Mask used to make the patches isotropic

Mask has been rescaled to contain 0 to 1 values so that it can be displayed using a gray color mask

```
filter = fspecial('gaussian', [9, 9], 2);
minf = min(min(filter));
maxf = max(max(filter));
filter = double(filter-minf)/double(maxf-minf);
filter = imresize(filter, [256, 256]);
figure;
imshow(uint8(255*filter));
colorbar;

toc;
```

```
Elapsed time is 126.366856 seconds.
```



Published with MATLAB® R2015b