Lab 3

CPS592 – Visual Computing and Mixed Reality

Preparation

- Open MATLAB
- Create Lab3 folder
- Copy *mountain.jpg* and *zbuilding.jpg* to Lab3 folder

Create script file

- Create Lab3.m inside Lab3 folder

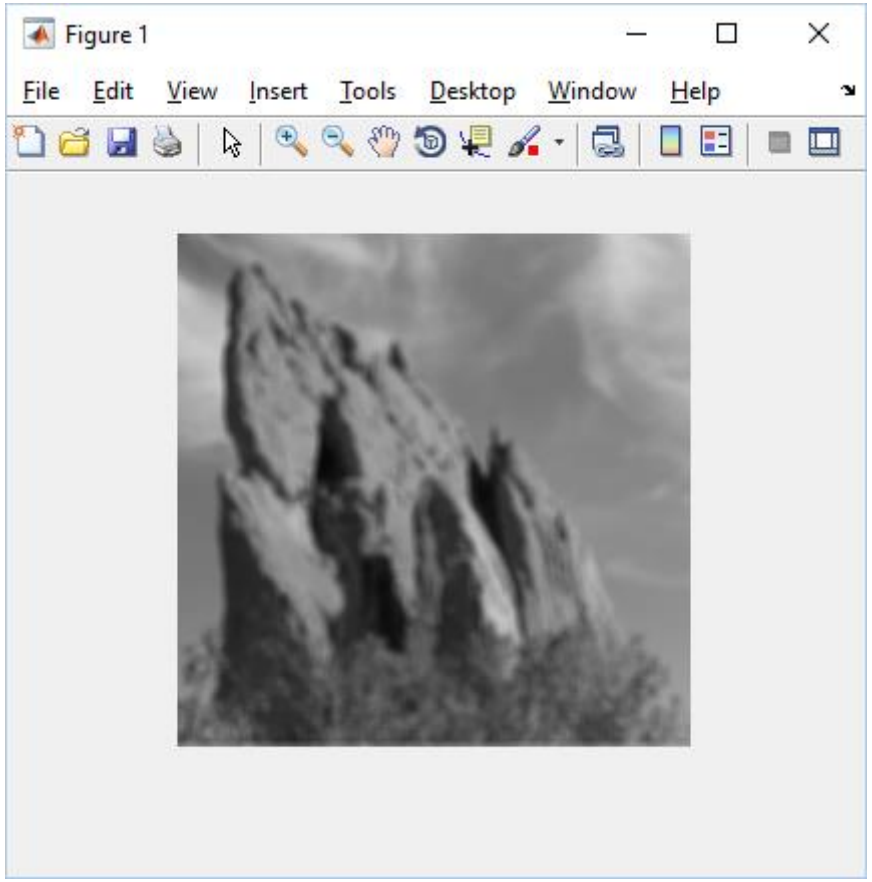
Clear previous data

```
%On top of lab3.m
close all;
clear all;
clc;
```

Load the image and apply Gaussian filter

```
img = imread('mountain.jpg');
img_gray = rgb2gray(img);
kernel_size = 5;
gaussian_kernel = fspecial('gaussian', [kernel_size kernel_size], 5);
img_gray_gaussian = imfilter(img_gray, gaussian_kernel, 'replicate');
figure, imshow(img_gray_gaussian);
```

Run script



Implement Bilateral Filter

```
% Preparation for BF
indent = (kernel_size - 1)/2;
[height, width] = size(img_gray);
img_results = zeros(height,width);
img_gray = double(img_gray);
sigma_range = 5;
```

Bilateral Filter with brute-force approach

```
% run nested loop for all image pixels
for i = indent + 1:height - indent
    for j = indent + 1:width - indent


end
end
```

Bilateral Filter with brute-force approach

```
% Compute range kernel
for i = indent + 1:height - indent
    for j = indent + 1:width - indent
        range_kernel = exp(-abs(img_gray(i - indent:i + indent,j - indent:j + indent )-
img_gray(i,j)).^2/(sigma_range * sigma_range));
end
end
```

Bilateral Filter with brute-force approach

```
% Compute joint kernel
for i = indent + 1:height - indent
    for j = indent + 1:width - indent
        range_kernel = exp(-abs(img_gray(i - indent:i + indent,j - indent:j + indent )-
img_gray(i,j)).^2/(sigma_range * sigma_range));
        kernel = range_kernel .* gaussian_kernel;
end
end
```

Bilateral Filter with brute-force approach

```
% Compute normalization factor
for i = indent + 1:height - indent
    for j = indent + 1:width - indent
        range_kernel = exp(-abs(img_gray(i - indent:i + indent,j - indent:j + indent )-
img_gray(i,j)).^2/(sigma_range * sigma_range));
        kernel = range_kernel .* gaussian_kernel;
        normalization = 1/sum(kernel(:));
end
end
```

Bilateral Filter with brute-force approach

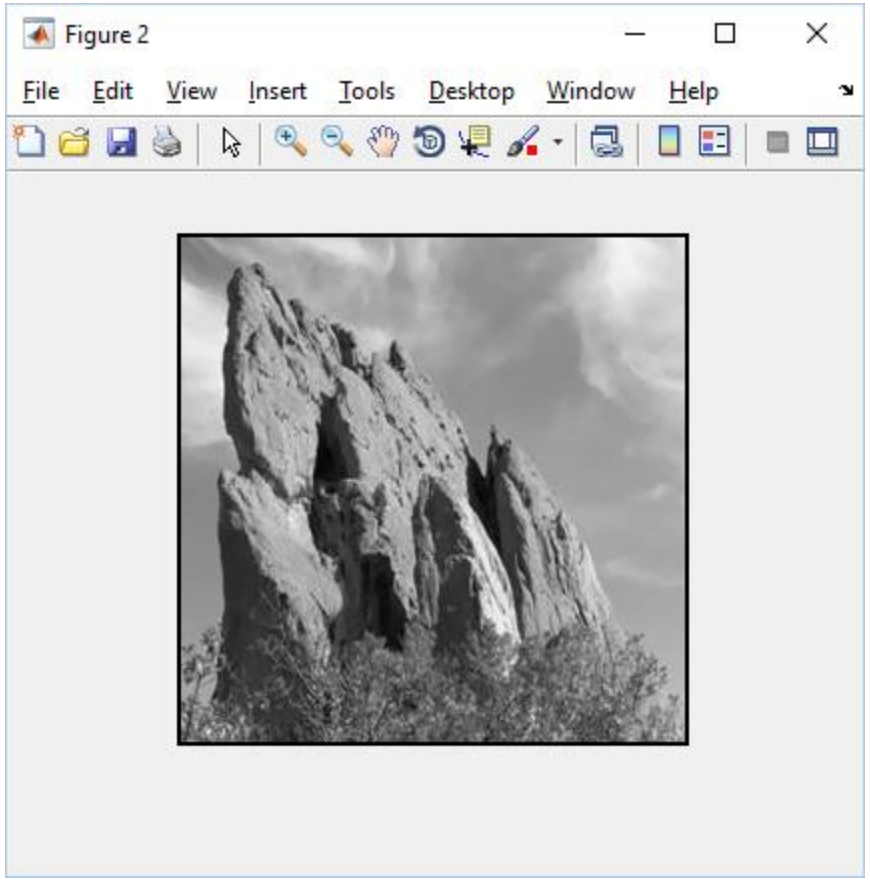
```
% Update the result to the output image
for i = indent + 1:height - indent
    for j = indent + 1:width - indent
        range_kernel = exp(-abs(img_gray(i - indent:i + indent,j - indent:j + indent )-
img_gray(i,j)).^2/(sigma_range * sigma_range));
        kernel = range_kernel .* gaussian_kernel;
        normalization = 1/sum(kernel(:));
        temp = (kernel.*double(img_gray(i - indent:i + indent,j - indent:j + indent))) *
normalization;
        img_results(i,j) = sum(temp(:));
end
end
```

Bilateral Filter with brute-force approach

```
% Display the result
for i = indent + 1:height - indent
   for j = indent + 1:width - indent
      range_kernel = exp(-abs(img_gray(i - indent:i + indent,j - indent:j + indent )-
img_gray(i,j)).^2/(sigma_range * sigma_range));
      kernel = range_kernel .* gaussian_kernel;
      normalization = 1/sum(kernel(:));
       temp = (kernel.*double(img_gray(i - indent:i + indent,j - indent:j + indent))) * normalization;
      img_results(i,j) = sum(temp(:));
    end
end
figure, imshow(img_results,[]);
```

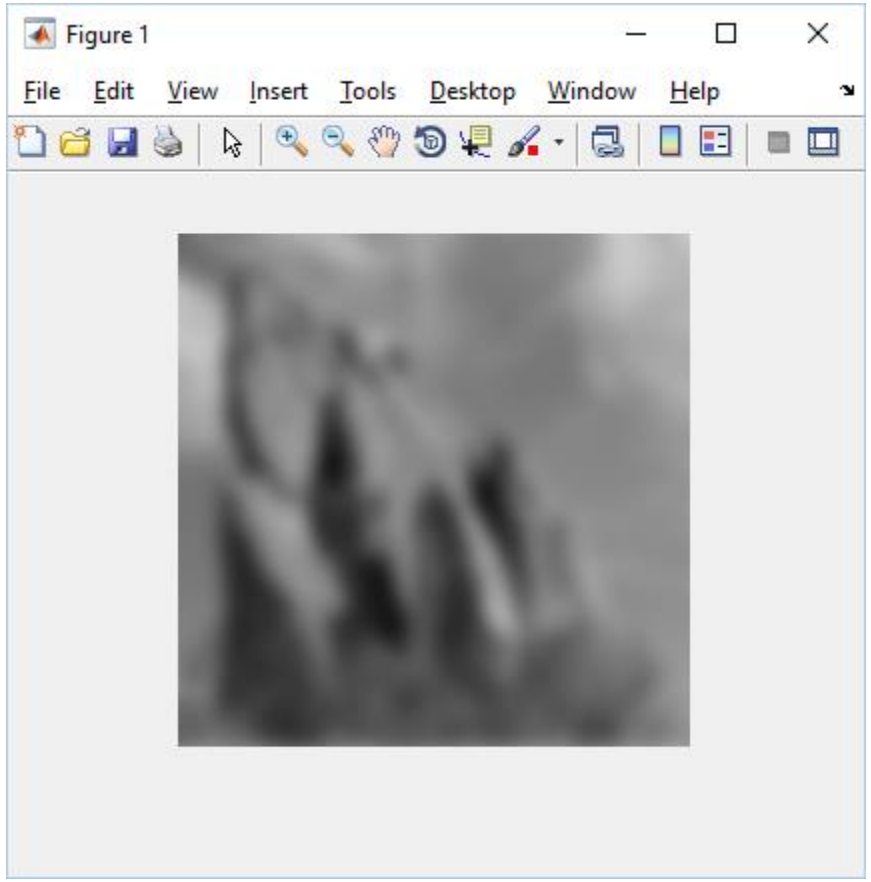
Run script and observe the results

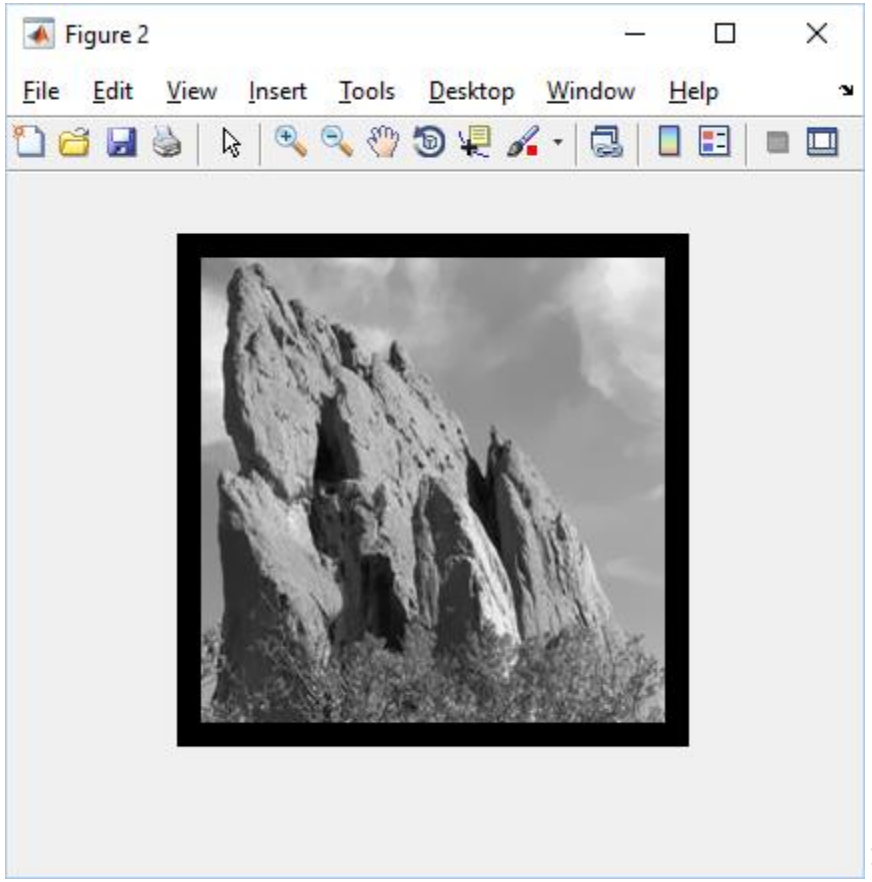
- Any observations?



Vary the kernel size

kernel_size = 25;

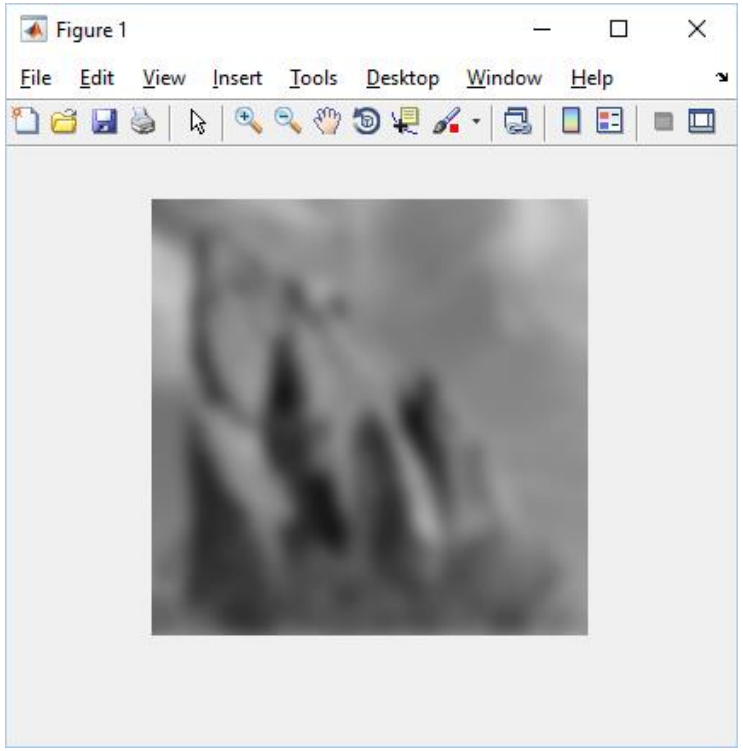


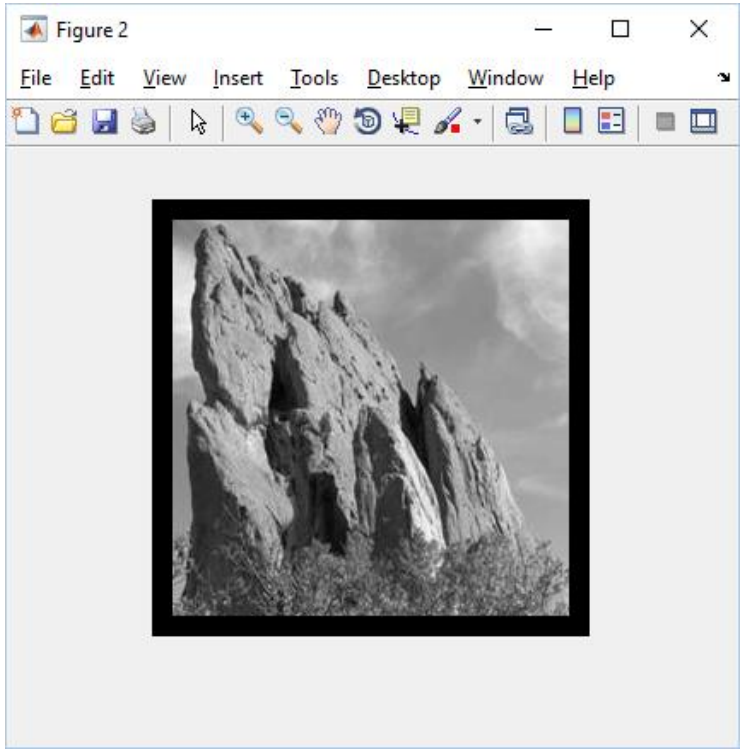


Vary the sigma_range

kernel_size = 25;

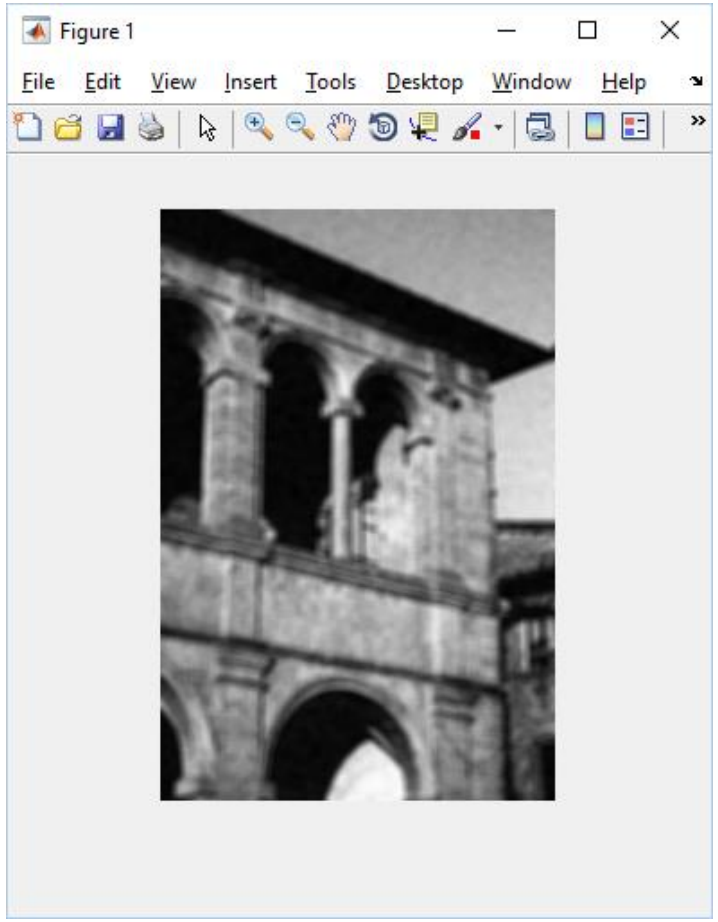
sigma_range = 0.1;

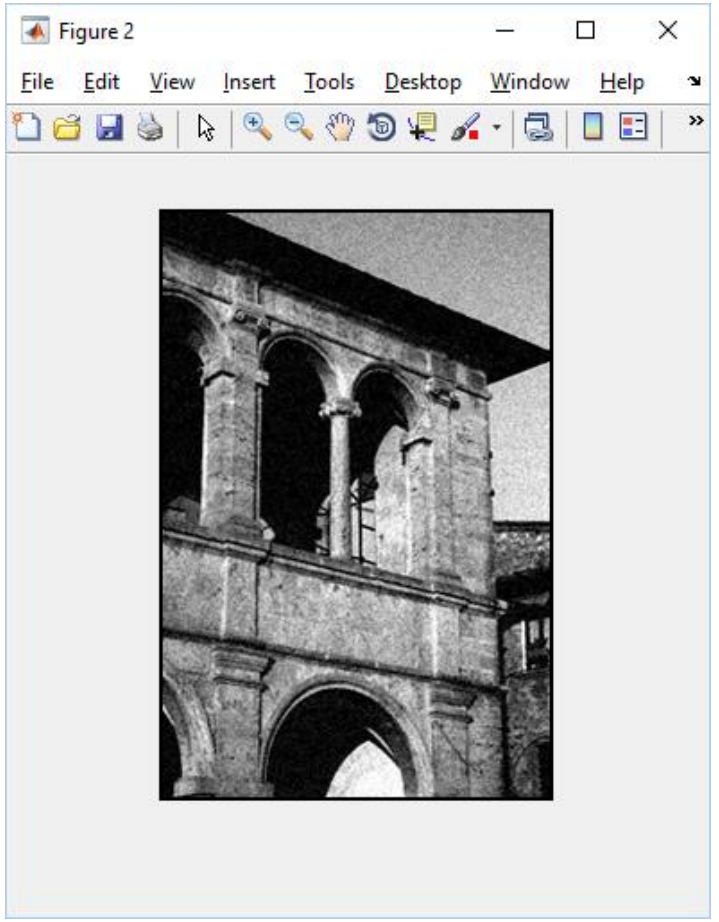




Try with different image

img = imread(zbuilding.jpg');





Q&A