Lab #2

Image Filtering (35)

In this lab, we will load an image and apply mean, Gaussian filters and median filters.

Part 1) **(5)**

I. Load the image *saphira.jpg* using the MATLAB function `imread`.

II. View the image using `imshow` (use `figure` to open a new figure window).

III. Create a mean filter kernel with a size of 31 pixels using `fspecial`.

```
kernelMean = fspecial('average',31);
```

IV. Apply the mean filter to the image using `imfilter`.

```
imgMean = imfilter(img,kernelMean);
```

V. Show the original and mean-filtered images in subplots in a figure window.

VI. Create a Gaussian filter kernel with a size of 31 pixels and standard deviation of 5 pixels.

```
kernelGaussian = fspecial('gaussian',31,5);
```

VII. Open a new figure window and use `surf` to view the Gaussian filter kernel shape.

```
surf(kernelGaussian);
```

VIII. Apply the Gaussian filter to the image and show the original and filtered images in subplots in a figure window.

IX. Add noise to the image using `imnoise`. Briefly describe the changes to the image you observe and how the `imnoise` function altered the image in computer memory.

```
imgNoisy = imnoise('salt & pepper',0.1);
```

X. Apply mean filtering to the noisy image using a filter of size 3.

XI. Apply Gaussian filtering to the noisy image using a filter of size 3 and standard deviation of 0.5.

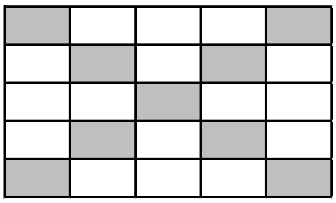
XII. Use the function `medfilt2` to apply median filtering to the image (default size of 3). Note that this needs to be done to individual channels one-at-a-time. First we copy the noisy image so that we can fill each channel with the filtered results.

```
imgNoisyMedian = imgNoisy;
imgNoisyMedian(:,:,1) = medfilt2(imgNoisy(:,:,1));
imgNoisyMedian(:,:,2) = medfilt2(imgNoisy(:,:,2));
imgNoisyMedian(:,:,3) = medfilt2(imgNoisy(:,:,3));
```

XIII. Use subplots in a new figure window to show and compare the different filtering results.

Part 2) **(10)**

Write a program in MATLAB that performs convolution on an image takes the average of the values on the diagonal of a 5 x 5 filter. (Grey cells as shown below are the cells to take the average over) Choose an image and convolve with your program. Comment on how the image turned out. Note that you should be taking the average over 9 cells. Hint: 1/9



Sample Calculation:

0.111	0	0	0	0.111
0	0.111	0	0.111	0
0	0	0.111	0	0
0	0.111	0	0.111	0
0.111	0	0	0	0.111

*

2	3	4	1	2	1
0	2	4	5	6	2
4	2	1	2	4	4
4	2	1	0	1	2
2	2	1	0	4	5
1	2	1	0	2	1

Red calculation:

2/9+ 2/9 + 2/9 + 5/9 + 1/9 + 2/9 + 0/9 + 2/9 + 4/9 = 2.22222

Green calculation:

3/9 + 1/9 + 1/9 + 4/9 + 6/9 + 2/9 + 1/9 + 1/9 + 2/9 + 5/9 = 2.77777

Part 3) **(20)**

Develop a well tuned and robust edge detection algorithm to consistently identify an object in scene of different light and backgrounds.