Eastern Mediterranean University

Department of Computer Engineering

CMPE/CMSE 424 Introduction to Image Processing

Laboratory Work #4

IMAGE RESTORATION AND FILTERING

The objective of this experiment is to implement noise models involved in the image restoration process and restore a degraded image by reducing noise using spatial filtering techniques. Noise models and filtering techniques can be implemented using MATLAB Image Processing Toolbox. Image restoration is directly related to reduce the noise from the degraded images. In this respect, the basic noise models namely Gaussian, Rayleigh, Gamma, Exponential, Uniform, Salt & Pepper noise will be studied.

EXPERIMENTAL WORK:

E1. Image restoration improves a given image in some predefined sense by modeling the degradation and applying the inverse process in order to recover the original image. In this respect, simulating the behaviour and effects of noise is important for image restoration process. Adding noise to an image can be done using MATLAB Image Processing Toolbox with "imnoise" and "imnoise2" functions [from the reference book "R. C. Gonzalez, R. E. Woods and S. L. Eddins, *Digital Image Processing using MATLAB, 2nd* Edition, Prentice Hall, 2009"**].**

(a) Use the predefined "imnoise" function and add Gaussian noise and Salt & Pepper noise to the image in Fig1.tif.

(b) Try to add other noise types included in "imnoise" function to the image in Fig1.tif and comment on the differences between these noise types.

(c) It is necessary to generate noise of types and parameters other than those available in function "imnoise". While "imnoise" outputs a noisy image, the function "imnoise2" **[**from the reference book "R. C. Gonzalez, R. E. Woods and S. L. Eddins, *Digital Image Processing using MATLAB, 2nd* Edition, Prentice Hall, 2009"**]** produces the noise pattern itself. The user may specify the desired values for

the noise parameters directly. Add the following noise types separately to the image in Fig1.tif using the function "imnoise2" :

(i) Uniform noise
(ii) Gaussian noise
(iii) Salt & Pepper noise
(iv) Lognormal noise (v) Rayleigh noise (vi) Exponential noise
(vii) Erlang noise

(d) Plot the histograms for each noise type generated in E1(c). The following code can be used to plot the histogram of Gaussian noise:

```
>> r=imnoise2('gaussian ',100000,1,0,1); % generates a column vector r, with 100000
% elements, each being a random number from a Gaussian distribution with mean 0
% and standard deviation of 1.
>> p=hist(r,50) % histogram of r is obtained
```

E2. When only noise is included for the degradation, spatial filtering can be used for noise reduction. The function "spfilt" is used to perform linear and nonlinear spatial filtering in the spatial domain with any of the filters namely Arithmetic mean, Geometric mean, Harmonic mean, Contraharmonic mean, Median, Max, Min, Midpoint, Alpha-trimmed mean.

(a) Use the image in Fig2.tif and the following commands to corrupt the image by pepper noise only with probability 0.1:

```
>> f=imread('Fig2.tif');
>> imshow(f)
>> [M, N]= size(f)
>> R = imnoise2('salt & pepper', M, N, 0.1, 0);
>> c = find (R==0);
>> gp = f;
>> gp(c) = 0;
>> imshow(gp)
```

(b) Use the image in Fig2.tif and the following commands to corrupt the image by salt noise only with probability 0.1:

```
>> f=imread('Fig2.tif');
```

```
>> imshow(f)
>> [M, N]= size(f);
>> R = imnoise2('salt & pepper', M, N, 0, 0.1);
>> c = find (R==1);
>> gs = f;
>> gs(c) = 255
>> imshow(gs)
```

(c) A good approach for filtering pepper noise is to use a contraharmonic filter with a positive value of Q. Apply 3x3 contraharmonic filter of order Q=1.5 on the image obtained in E2 (a) and display the result of filtering using the following command:

```
>> fp = spfilt(gp, 'chmean ', 3, 3, 1.5);
>> imshow(fp)
```

(d) Salt noise can be filtered using a contraharmonic filter with a negative value of Q. Apply 3x3 contraharmonic filter of order Q=-1.5 on the image obtained in E2 (b) and display the result of filtering using the following command:

```
>> fs = spfilt(gs, 'chmean ', 3, 3, -1.5);
>> imshow(fs)
```

(e) Use the following commands for filtering pepper noise with a 3x3 max filter on the image obtained in E2 (a) and display the result of filtering using the following command

```
>> fpmax = spfilt(gp, 'max ', 3, 3);
>> imshow(fpmax)
```

(f) Use the following commands for filtering salt noise with a 3x3 min filter on the image obtained in E2 (b) and display the result of filtering using the following command

```
>> fsmin = spfilt(gs, 'min ', 3, 3);
>> imshow(fsmin)
```

E3. Write down a <u>report</u> (2 or 3 pages) discussing the results obtained in E1 and E2. The report will be submitted to the assistant 1 day after the lab date. ***(Discussion of the results must be written with your own words, otherwise the report will not be graded.)***