영상처리 HW #3

Due on 20 May

1. Edge detection operators can be compared in an objective way. The performance of an edge detection operator in noise can be measured quantitatively as follows: **Let n_0 be the number of edge pixels declared** and **n_1 be number of missed or new edge pixels after adding noise**. If n_0 is held fixed for the noiseless as well as noisy images, then the **edge detection error rate (Pe)** is
(n0 = 변화된 edge pixel의 수, n1 = edge에 해당하는 픽셀수)

$$P_e = \frac{n_1}{n_0}.$$

Compare the performance of the gradient operators of Roberts, Sobel, Prewitt and the 5x5 stochastic gradient on a noisy image with **SNR= 8dB**.
Note that the pixel location (*m*,*n*) is declared an edge location if the magnitude gradient $g(m,n) = \sqrt{g_x^2(m,n) + g_y^2(m,n)}$ exceeds a **THRESH value of 150**. The edge locations constitute an *edge map*. For this assignment, you can select 512x512 BMP or RAW grayscale image of Lena.

2. Compare the performance between the 3x3 Low-pass and Median filters for a noisy image with SNR=9dB. For an objective comparison, obtain the MSE (mean square error) for each result. For this assignment, use 512x512 grayscale image of BOAT.raw.

(1) Method to generate a noisy image with Gaussian noise

The **signal-to-noise ratio (SNR)(잡음정도)** is expressed in decibels as

$$SNR = 10\log_{10}\frac{\sigma^2}{\sigma_e^2} \text{ (dB)}$$

where σ^2 is the variance of the original image and $\sigma_e{}^2$ is the variance of the noise signal. In order to generate a noisy image with a specified SNR by adding Normalized Gaussian Noise with distribution of $N(0,\sigma_e^2)$, you can refer to the following source codes.

```
------------------------------------------------------------------------------------------
variance=get_image_power(input_image);     /* The data type for input _image should
be Int */
stddev_noise=sqrt(variance/pow(10.0,((double) SNR/10));    /* Here SNR is set at 8 */
AddGaussianNoise(input_image, noise_image, stddev_noise);

AddGaussianNoise(input_img, noise_img, sigma)    /*Add Gaussian Noise to the input
image */
int input_img[][N], noise_img[][N];
double sigma;
{
   int i,j,s;
   for(i=0;i<N;i++)
      for(j=0;j<N; j++)
      {
          s=input_img[i][j]+Gaussian(sigma);
          noise_img[i][j]= s>255 ? 255 : s<0 ? 0 : s;
      }
}
float Gaussian(sd)
float sd;
{
```

```
static int ready =0;
static float gstore;
float v1, v2, r, fac, gaus;
int r1, r2;

if(ready==0) {
   do {
         r1=rand();
         r2=rand();
         v1=2.*((float)r1/(float) RAND_MAX-0.5);
         v2=2.*((float)r2/(float) RAND_MAX-0.5);
         r=v1*v1+v2*v2;
       } while (r>1.0);
   fac=(float) sqrt((double) (-2*log(r)/r));
   gstore=v1*fac;
   gaus=v2*fac;
   ready=1;
}
else {
   ready=0;
   gaus=gstore;
}
return (gaus*sd);
}
```

(2) Method to compute an image power or variance

The signal variance can be represented as follows:

$$\sigma^2 = E[(X-\eta)^2] = E[X^2 - 2X\eta + \eta^2] = E[X^2] - 2\eta E[X] + \eta^2 = E[X^2] - \eta^2$$

where $E[X] = \eta$.

Thus, the image power for *MXN* image can be obtained by using the following equation:

$$\sigma^2 = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} X_{ij}^2 - \left(\frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} X_{ij}\right)^2.$$

(3) Edge Detection Masks

1) As for the Roberts, Sobel, and Prewitt mask, please use the masks as shown in Fig. 3.16 on page 109 of the text book.

2) As for the 5x5 stochastic gradient mask, please use the mask shown below.

$$G_X = \begin{bmatrix} 0.267 & 0.364 & 0 & -0.364 & -0.267 \\ 0.373 & 0.562 & 0 & -0.562 & -0.373 \\ 0.463 & 1.000 & 0 & -1.000 & -0.463 \\ 0.373 & 0.562 & 0 & -0.562 & -0.373 \\ 0.267 & 0.364 & 0 & -0.364 & -0.267 \end{bmatrix}$$

$$G_Y \equiv G_X^T$$