

Computer and Robot Vision

Homework#8

R01944040 柳成蔭

這次的作業是對原圖增加 gaussian noise 和 salt-and-pepper noise，然後先別用 box filter, median filter, opening-then-closing and closing-then opening filter 去除 noise。

我使用 VS2012 編寫程式

(1) 對 Lena 圖增加 Gaussian 噪聲。 Amplitude 分別取 10 和 30。

$$I(nim, i, j) = I(im, i, j) + amplitude * N(0,1)$$

$N(0,1)$: Gaussian random variable with zero mean and st. dev. 1

```
void GaussianNoise(Mat noise, double amplitude)
{
    for(int i=0; i<=noise.rows-1; i++)
    {
        for(int j=0; j<=noise.cols-1; j++)
        {
            int
Inim=noise.at<uchar>(i,j)+(int)( amplitude*Normal() );
            if(Inim<=0)
                noise.at<uchar>(i,j)=0;
            else if(Inim>=255)
                noise.at<uchar>(i,j)=255;
            else
                noise.at<uchar>(i,j)=Inim;
        }
    }
}
```

并計算產生圖的 SNR。

```
double VS(const Mat src)
{
    int N=src.rows*src.cols;
    double u=0;
    for(int i=0; i<=src.rows-1; i++)
    {
        for(int j=0; j<=src.cols-1; j++)
        {
            u=u+src.at<uchar>(i,j);
        }
    }
    u=u/N;
    double vS=0;
    for(int i=0; i<=src.rows-1; i++)
    {
        for(int j=0; j<=src.cols-1; j++)
        {
            vS=vS+pow( (src.at<uchar>(i,j)-u), 2 );
        }
    }
    vS=vS/N;
    return vS;
}

double VN(const Mat src, const Mat noise)
{
    int N=src.rows*src.cols;
    double uN=0;
    for(int i=0; i<=src.rows-1; i++)
    {
        for(int j=0; j<=src.cols-1; j++)
        {
            uN=uN+( noise.at<uchar>(i,j) -
src.at<uchar>(i,j) );
        }
    }
    uN=uN/N;
}
```

```

double vN=0;
for(int i=0; i<=src.rows-1; i++)
{
    for(int j=0; j<=src.cols-1; j++)
    {
        vN=vN+pow( (noise.at<uchar>(i,j) -
src.at<uchar>(i,j) -uN), 2 );
    }
}
vN=vN/N;
return vN;
}

double SNR(const Mat src, const Mat noise)
{
    double vS;
    vS=VS(src);
    double vN;
    vN=VN(src, noise);
    double snr;
    snr=20*log10(sqrt(vS)/sqrt(vN));
    return snr;
}

```

Gaussian Noise amplitude 10 & Gaussian Noise amplitude 30:



Gaussian Noise amplitude 10 的 SNR 為 13.9416 。

Gaussian Noise amplitude 30 的 SNR 為 4.2699 。

(2) 對 Lena 圖增加 Salt-and-Pepper 噪聲。 Threshold 分別取 0.05 和 0.1。

$$I(\text{noiseImage}, i, j) = \begin{cases} 0 & , \text{if } \text{uniform}(0,1) < 0.05 \\ 255 & , \text{if } \text{uniform}(0,1) > 1 - 0.05 \\ I(\text{srcImage}, i, j) & , \text{otherwise} \end{cases}$$

uniform(0,1): random variable uniformly distributed over [0,1]

```
void SaltPepperNoise(Mat noise, double threshold)
{
    for(int i=0; i<=noise.rows-1; i++)
    {
        for(int j=0; j<=noise.cols-1; j++)
        {
            double u=Uniform();
            if(u<threshold)
                noise.at<uchar>(i,j)=0;
            else if(u>1-threshold)
                noise.at<uchar>(i,j)=255;
            else
                noise.at<uchar>(i,j)=noise.at<uchar>(i,j);
        }
    }
}
```

Salt-and-Pepper Noise Threshold 0.05 & Salt-and-Pepper Noise Threshold 0.1 :



Salt-and-Pepper Noise Threshold 0.05 的 SNR 為 0.8780。

Salt-and-Pepper Noise Threshold 0.1 的 SNR 為 -2.0969。

(a) Box filter

box filter: computes equally weighted average

box filter: separable

box filter: recursive implementation with “two+”, “two-”, “one/”
per pixel

1	1	1
1	1	1
1	1	1

(a) $\frac{1}{9}$

1	1	1
1	2	1
1	1	1

(b) $\frac{1}{10}$

1	2	1
2	4	2
1	2	1

(c) $\frac{1}{16}$

```
void BoxFilter(const Mat src, Mat res, int size)
{
    int sum=0;
    int n;
    for(int i=0; i<=res.rows-1; i++)
    {
        for(int j=0; j<=res.cols-1; j++)
        {
            sum=0;
            n=0;
            for(int x=i-size/2; x<=i+size/2; x++)
            {
                if(x>=0 && x<=res.rows-1)
                {
                    for(int y=j-size/2; y<=j+size/2; y++)
                    {
                        if(y>=0 && y<=res.cols-1)
                        {
                            sum=sum+src.at<uchar>(x,y);
                            n=n+1;
                        }
                        else
                            continue;
                    }
                }
            }
        }
    }
}
```



```

        }
        else
            continue;
    }
    res.at<uchar>(i,j)=(int)(sum/n);
}
}
}

```

3*3 的 Box filter 對 Gaussian Noise amplitude 10 的處理結果 & 對 Gaussian Noise amplitude 30 的處理結果：



3*3 的 Box filter 對 Gaussian Noise amplitude 10 處理的 SNR 為 17.8309。

3*3 的 Box filter 對 Gaussian Noise amplitude 30 處理的 SNR 為 12.7173。

5*5 的 Box filter 對 Gaussian Noise amplitude 10 的處理結果 & 對 Gaussian Noise amplitude 30 的處理結果：



5*5 的 Box filter 對 Gaussian Noise amplitude 10 處理的 SNR 為 14.8834。

5*5 的 Box filter 對 Gaussian Noise amplitude 30 處理的 SNR 為 13.3686。

3*3 的 Box filter 對 Salt-and-Pepper Noise Threshold 0.05 的處理結果 & 對 Salt-and-Pepper Noise Threshold 0.1 的處理結果：



3*3 的 Box filter 對 Salt-and-Pepper Noise Threshold 0.05 處理的 SNR 為 9.3992。

3*3 的 Box filter 對 Salt-and-Pepper Noise Threshold 0.1 處理的 SNR 為 6.3556。

5*5 的 Box filter 對 Salt-and-Pepper Noise Threshold 0.05 的處理結果 & 對 Salt-and-Pepper Noise Threshold 0.1 的處理結果：



5*5 的 Box filter 對 Salt-and-Pepper Noise Threshold 0.05 處理的 SNR 為 11.0981。

5*5 的 Box filter 對 Salt-and-Pepper Noise Threshold 0.1 處理的 SNR 為 8.5370。

(b) Median filter

median: effective **for impulsive noise** (salt and pepper)

median: distorts or loses fine detail such as thin lines

$$Z_{median} = X_{(\frac{N+1}{2})}$$

```
void MedianFilter(const Mat src, Mat res, int size)
{
    for(int i=0; i<=res.rows-1; i++)
    {
        for(int j=0; j<=res.cols-1; j++)
        {
            vector<int> pixelValue;
            for(int x=i-size/2; x<=i+size/2; x++)
            {
                if(x>=0 && x<=res.rows-1)
                {
                    for(int y=j-size/2; y<=j+size/2; y++)
                    {
                        if(y>=0 && y<=res.cols-1)
                        {
                            pixelValue.push_back( src.at<uchar>(x,y) );
                        }
                        else
                            continue;
                    }
                }
                else
                    continue;
            }
            sort(pixelValue.begin(),pixelValue.end());
            int N=pixelValue.size();
            res.at<uchar>(i,j)=pixelValue[(N+1)/2-1];
        }
    }
}
```


}

3*3 的 Median filter 對 Gaussian Noise amplitude 10 的處理結果 & 對 Gaussian Noise amplitude 30 的處理結果：



3*3 的 Median filter 對 Gaussian Noise amplitude 10 處理的 SNR 為 17.9289。

3*3 的 Median filter 對 Gaussian Noise amplitude 30 處理的 SNR 為 11.2083。

5*5 的 Median filter 對 Gaussian Noise amplitude 10 的處理結果 & 對 Gaussian Noise amplitude 30 的處理結果：



5*5 的 Median filter 對 Gaussian Noise amplitude 10 處理的 SNR 為 16.0848。

5*5 的 Median filter 對 Gaussian Noise amplitude 30 處理的 SNR 為 13.0295。

3*3 的 Median filter 對 Salt-and-Pepper Noise Threshold 0.05 的處理結果 &

對 Salt-and-Pepper Noise Threshold 0.1 的處理結果：



3*3 的 Median filter 對 Salt-and-Pepper Noise Threshold 0.05 處理的 SNR 為 19.2650。

3*3 的 Median filter 對 Salt-and-Pepper Noise Threshold 0.1 處理的 SNR 為 15.0678。

5*5 的 Median filter 對 Salt-and-Pepper Noise Threshold 0.05 的處理結果 & 對 Salt-and-Pepper Noise Threshold 0.1 的處理結果：



5*5 的 Median filter 對 Salt-and-Pepper Noise Threshold 0.05 處理的 SNR 為 16.3209。

5*5 的 Median filter 對 Salt-and-Pepper Noise Threshold 0.1 處理的 SNR 為 15.8586。

(c) Opening-then-closing filter

先設置 morphology 使用的 kernel，dilation 和 erosion 使用 octogonal 3-5-5-5-3 的 kernel。

```
struct Kernel
```

```

{
    int kCols;
    int kRows;
    int anchorX;
    int anchorY;
    Mat values;
    Kernel(int cols, int rows, int ancx, int ancy, Mat val)
        :kCols(cols), kRows(rows), anchorX(ancx),
        anchorY(ancy), values(val.clone())
    {
    }
};

uchar kValArr[]={0, 255,255,255,0,
                  255,255,255,255,255,
                  255,255,255,255,255,
                  255,255,255,255,255,
                  0, 255,255,255,0 };
Mat kVal=Mat(5,5,CV_8U,kValArr).clone();
Kernel ker(5, 5, 2, 2, kVal);

```

- gray scale opening of f by kernel k denoted by $f \circ k$ $f \circ k = (f \ominus k) \oplus k$

- gray scale closing of f by kernel k denoted by $f \bullet k$

$$f \bullet k = (f \oplus k) \ominus k$$

Opening會先對影像進行一次erosion，然後對得到的結果進行一次dilation。

Closing則相反，先對影像進行一次dilation，然後對得到的結果進行一次erosion。

例如對 Gaussian Noise amplitude 10 的圖做 opening-then-closing filter 就是對圖做一次 erosion 再兩次 dilation 再一次 erosion。

```

Mat imgErosionGaussian10(imgGaussian10.cols,
imgGaussian10.rows, CV_8U, Scalar(0) );

```

```

    grayErosion( imgGaussian10, imgErosionGaussian10,
ker );
    Mat imgOpenGaussian10(imgGaussian10.cols,
imgGaussian10.rows, CV_8U, Scalar(0) );
    grayDilation( imgErosionGaussian10, imgOpenGaussian10,
ker );
    Mat imgOpenDilationGaussian10(imgGaussian10.cols,
imgGaussian10.rows, CV_8U, Scalar(0) );
    grayDilation( imgOpenGaussian10,
imgOpenDilationGaussian10, ker );
    Mat imgOpenCloseGaussian10(imgGaussian10.cols,
imgGaussian10.rows, CV_8U, Scalar(0) );
    grayErosion( imgOpenDilationGaussian10,
imgOpenCloseGaussian10, ker );

```

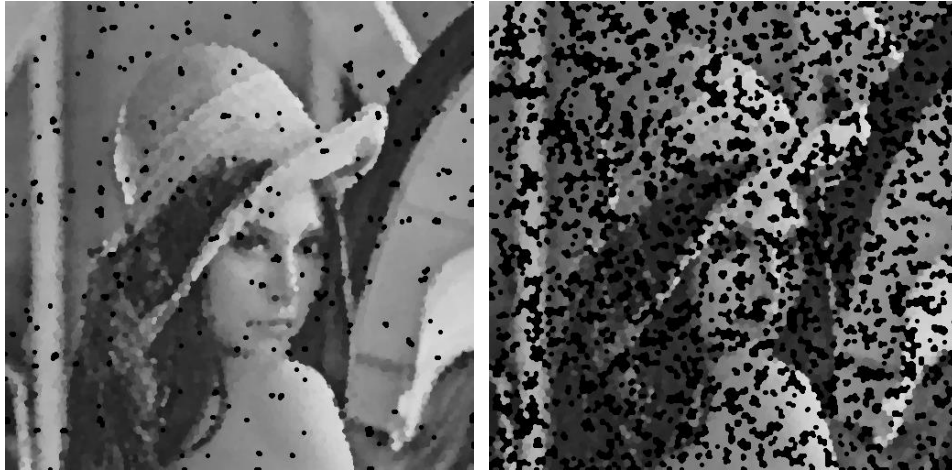
Opening-then-closing filter 對 Gaussian Noise amplitude 10 的處理結果 & 對 Gaussian Noise amplitude 30 的處理結果：



Opening-then-closing filter 對 Gaussian Noise amplitude 10 處理的 SNR 為 13.2973 。

Opening-then-closing filter 對 Gaussian Noise amplitude 30 處理的 SNR 為 11.1749 。

Opening-then-closing filter 對 Salt-and-Pepper Noise Threshold 0.05 的處理結果 & 對 Salt-and-Pepper Noise Threshold 0.1 的處理結果：



Opening-then-closing filter 對 Salt-and-Pepper Noise Threshold 0.05 處理的 SNR 為 5.7309。

Opening-then-closing filter 對 Salt-and-Pepper Noise Threshold 0.1 處理的 SNR 為 -2.2214。

(d) Closing-then-opening filter

例如對 Gaussian Noise amplitude 10 的圖做 closing-then-opening filter 就是對圖做一次 dilation 再兩次 erosion 再一次 dilation。

```
Mat imgDilationGaussian10(imgGaussian10.cols,
imgGaussian10.rows, CV_8U, Scalar(0) );
grayDilation( imgGaussian10, imgDilationGaussian10,
ker );
Mat imgCloseGaussian10(imgGaussian10.cols,
imgGaussian10.rows, CV_8U, Scalar(0) );
grayErosion( imgDilationGaussian10, imgCloseGaussian10,
ker );
Mat imgCloseErosionGaussian10(imgGaussian10.cols,
imgGaussian10.rows, CV_8U, Scalar(0) );
grayErosion( imgCloseGaussian10,
imgCloseErosionGaussian10, ker );
Mat imgCloseOpenGaussian10(imgGaussian10.cols,
imgGaussian10.rows, CV_8U, Scalar(0) );
grayDilation( imgCloseErosionGaussian10,
imgCloseOpenGaussian10, ker );
```


Closing-then-opening filter 對 Gaussian Noise amplitude 10 的處理結果 & 對 Gaussian Noise amplitude 30 的處理結果：



Closing-then-opening filter 對 Gaussian Noise amplitude 10 處理的 SNR 為 13.5960。

Closing-then-opening filter 對 Gaussian Noise amplitude 30 處理的 SNR 為 11.1316。

Closing-then-opening filter 對 Salt-and-Pepper Noise Threshold 0.05 的處理結果 & 對 Salt-and-Pepper Noise Threshold 0.1 的處理結果：



Closing-then-opening filter 對 Salt-and-Pepper Noise Threshold 0.05 處理的 SNR 為 5.0632。

Closing-then-opening filter 對 Salt-and-Pepper Noise Threshold 0.1 處理的 SNR 為 -2.5950。