**Computer and Robot Vision**

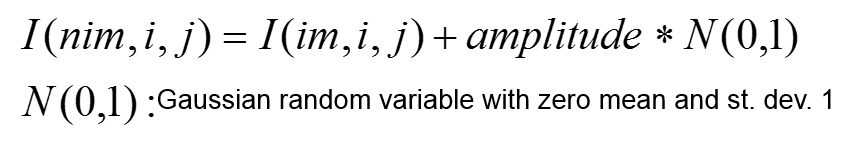
**Homework#8**

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。



|  |
| --- |
| 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;  } |

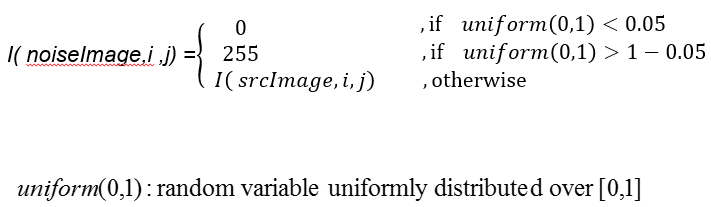
Gaussian Noise amplitude 10 & Gaussian Noise amplitude 30：

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。



|  |
| --- |
| 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);  }  }  } |

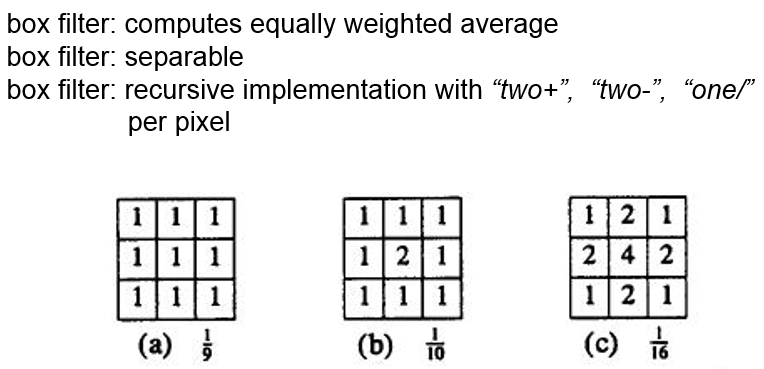
Salt-and-Pepper Noise Threshold 0.05 & Salt-and-Pepper Noise Threshold 0.1：

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

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

1. Box filter



|  |
| --- |
| 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。

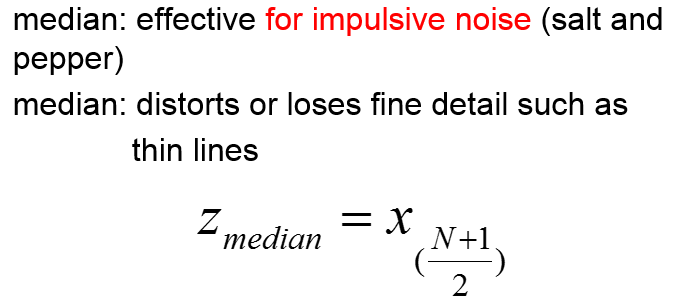
5\*5的Box filter對Salt-and-Pepper Noise Threshold 0.05的處理結果 & 對Salt-and-Pepper Noise Threshold 0.1的處理結果：

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。

1. Median filter



|  |
| --- |
| 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：

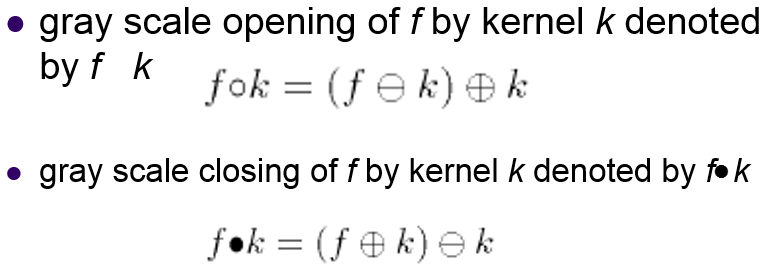
5\*5的Median filter對Salt-and-Pepper Noise Threshold 0.05處理的SNR為16.3209。

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

1. 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); |



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。

1. 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。