# Image Enhancement Matlab Code

Abstract – This report documents noise removal and colour enhancement image processing techniques. Different techniques are applied and analysed to deduce a combination of image enhancement filters. Details of experimentation and a conclusion are detailed in the report.

## Introduction

## Methodology

Solution 1:

* Read image
* Apply median filtering. Removes salt and pepper noise
* Fast Fourier transform image
* Shift fast Fourier transformed image into new matrix

Solution 2:

* Read image
* Fast Fourier transform image
* Shift fast Fourier transformed image into new matrix
* Display magnitude of Fourier transformation. Explain.
* Display phase of Fourier transformation. Explain.
* Process image. Noise removal
  + Low-pass filtering
* Display magnitude and phase of image after filtering.
* Inverse Fourier transformation.
* Display final image.

## Results

Fig.0 Magnitude

Fig.0 Phase

## Discussion

Converting an image from its true colour representation (RGB) to greyscale converts the 3D matrix into a single 2D matrix of which each element is an intensity value. These values are indexed to a colour map.

## Conclusion

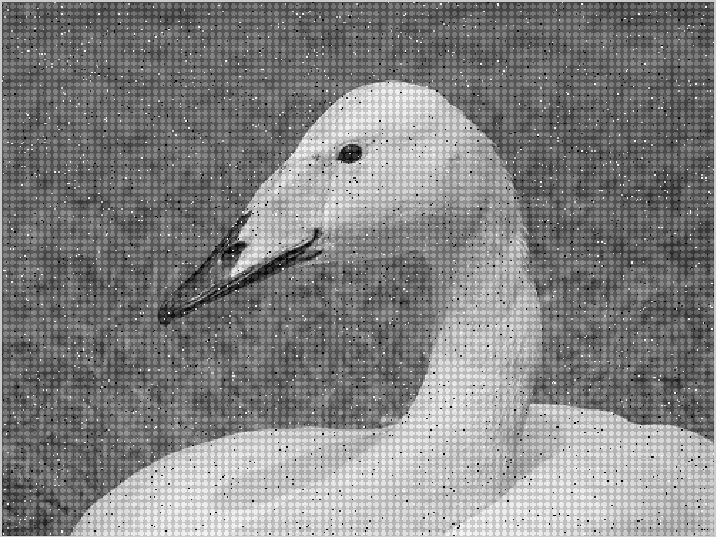
## References

## Appendix

// read in image to new variable originalImage

OriginalImage = Imread(‘filePath.fileName.fileExtension’)

Imshow(I)



// if image is not already grey convert 3-D array into 2-D array and greyscale else assign

[ rows columns numberOfColorChannels ] = size(I);

If() numberOfColorChannels > 1

ImageGray = rgb2gray(I);

else

ImageGray = I; // image is already gray

End

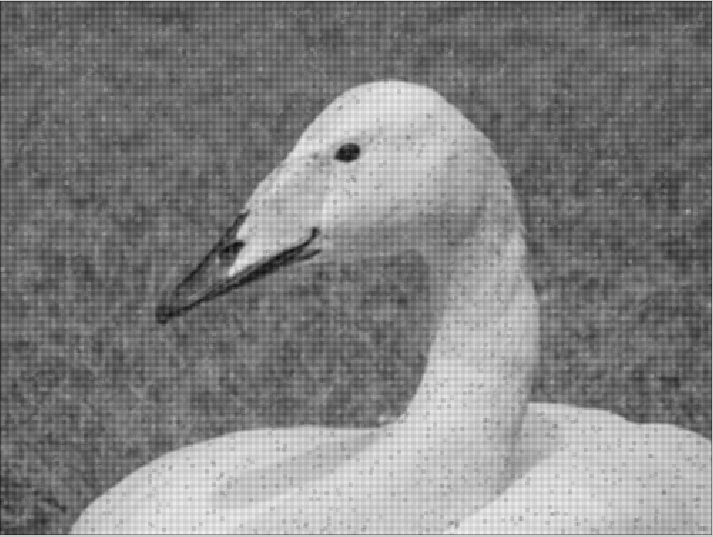
// Linear filtering

// Input grey scale image

// averaging filter to remove random (salt and pepper) noise

AvergingFilterImage = filter2(fspecial('average',3),ImageGray)/255;

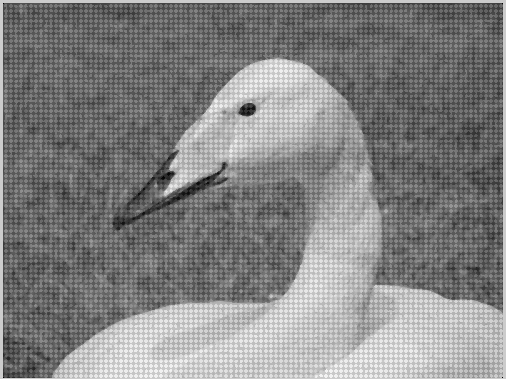
imshow(AvergingFilterImage)



// Median filter removal

MedianFilteredImage = medfilt2(ImageGray,[3 3]);

imshow(MedianFilteredImage)

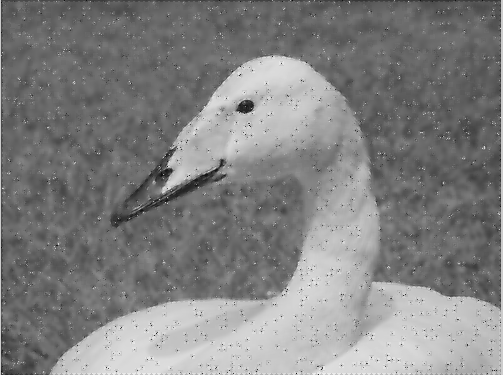


// Adaptive Filtering

// Add Weiner filter

AdaptiveFilteredImage = wiener2(ImageGray,[5 5]);

Imshow(AdaptiveFilteredImage)



// Structured noise removal will require Fourier Transformation

**//Program Code 1**

/load Original grey image

// transform image into frequency domain

//display magnitude and phase of image

//create kernel

//convolute shifted Fourier and kernel matrices

//show filtered image

I = imread('\\ndrive\xw009807\.do\_not\_delete\desktop.xp\IA assignment\Images\swanNoise.bmp');

originalfft = fft2(I)

shiftedfft = fftshift()

imshow(log(abs(shifted),[]))

rectWidth = 10

rectHeight = 5

kernel = ones(rectWidth, rectHeight) / (rectHeight \* rectWidth)

filteredImage = conv2(double(I), kernel, ‘same’)

imshow(filteredImage, [])

References

* <http://uk.mathworks.com/help/images/noise-removal.html>