**Lab 1: Point Processing + Spatial Filtering + Frequency Filtering + Imaging Geometry**

**2.1 Contrast Stretching**

The imadjust function from the Image Processing Toolbox (IPT) can be utilized to do contrast stretching. It maps a defined input intensity range, minimum to maximum pixel intensity of the original image, into a defined output intensity range, 0 to 255 {18}\*. A requirement is that input and output must be normalized to be between 0 and 1 {16,17}.

Comparing the original and stretched image in fig. 1, the difference between light and dark spots are visually higher in the contrast stretched image. The histogram reflects the images, with the bright spike at higher intensity and the low spikes at lower intensity.

\*{} indicate the line number(s) in the source code [Refer to Appendix A]

Fig. 1. Result of Contrast Stretching on image, “mrt-train.jpg”.

**2.2 Histogram Equalization**

Contract can also be enhanced using histogram equalization, spreading out the most frequent intensity values, demonstrated by the difference between “Original” and “Equalized” results in fig. 2. This can be done using the histeq function from the IPT {26}. Repeated application of histogram equalization result in no changes as it is idempotent {27}.

Fig. 2. Result of Histogram Equalization on image, “mrt-train.jpg”.

**2.3 Linear Spatial Filtering**

Linear Spatial Filtering can be done convulating the image with a small spatial filter in the form of a 5x5 kernel. Kernel is displayed in fig. 3.

Fig. 3. 5x5 Kernel used for linear spatial filtering

As the sigma value in the filter equation, , increases, more noise is removed both for the gaussian noise and the speckle noise. However, the contrast of the image is reduced and it becomes visually more blur as a result as shown in fig. 4.

Linear Spacial Filtering is more effective on gaussian noise than speckle noise, this is visually noticible for sigma=2. The gaussian noise is mostly removed while the white speckles are still evident.

For code implementation, the kernel is created {48-55} and made into a averaging filter {56}. It is then convoluted with the images Gn and Sp {63-79}. The whole process is repeated for sigma = 2 {91-126}.

Fig. 4. Result of Linear Spatial Filtering on images with gaussian noise, “ntu-gn.jpg”, and speckle noise, “ntu-sp.jpg”.

**2.4 Median Filtering**

The Median Filter, as opposed to the gaussian averaging filter, is non-linear. Comparing the results in fig. 5, we can observe the following. In terms of noise reduction, it is also much more effective at removing speckle noise, with a simple 3x3 median filter being able to completely remove speckle noise from the image. However, there is no visible difference for the image with gaussian noise between the median filter and gaussian averaging filter.

The advantages of the median filter over the gaussian averaging filter are its simplicity and its ability to retain edges well. The tradeoff is the higher computational time required, due to sorting. The gaussian averaging filter, in comparison, uses only addition and multiplication, resulting in lower computational time.

Fig. 5. Result of Median Filtering on images with gaussian noise, “ntu-gn.jpg”, and speckle noise, “ntu-sp.jpg”.

**2.5.1 Suppressing Noise Interference Patterns for “pck-int.jpg”**

By editing the image in the Fourier domain, it allows for removal of high frequency noise while maintaining the image data that is stored at low frequency. From the “Final” image in fig. 6, most of the noise has been removed and the image beneath is much clearer. However, remnants of noise remain, especially at the edges of the image.

One possibility is due to the measurement of the noise frequency being inaccurate due to human error. The values used are solely based on human vision and decision on where the noise is {168-178}. A solution would be the segment the power spectrum into multiple frequency cubes and search for the range of noise frequencies using intensity of yellow light, removing them. The center segments can be excluded to prevent important data from being removed.

Fig. 6. Result of Noise Suppression on image, “pck-int.jpg”, and their respective power spectrum representations.

**2.5.2 Suppressing Noise Interference Patterns for “primate-caged.jpg”**

Applying the noise suppression technique from earlier on primate-caged.jpg, the “Final” result in fig. 7 was subpar as compared the pck-int.jpg image {191-222}. The cage still remained clearly visible despite removing the noise spots in the fourier domain {203-213}. Increasing the size of the filtering boxes will further remove the cage. However, it will remove important data of the monkey, causing it too look blur. This is due to the frequency of the noise and the frequency of the monkey image being similar, as exhibited by the image dot and noise dots being close to each other in the “Power Spectrum”.

Fig. 7. Result of Noise Suppression on image, “primate-caged.jpg”, and their respective power spectrum representations.

**2.6 Undoing Perspective Distortion of Planar Surface**

Undoing perspective distortion of a planar surface can be done by generating the 2D planar projective transform, and applying the transform to the image.

The coordinates of each edge of the book in “book.jpg” are found using ginput {235} and placed into column and row vectors {238,239}. The four fixed points, in the base vector, represents the proposed coordinates of each edge of the book {240}. Using fitgeotrans\* function, we are able to calculate the 3x3 projective transformation matrix, in table. 1, from the initial column and row vectors, and the base vector {242-246}.

\*The fitgeotrans function and the maketform function are interchangable, but it is recommended by the mathworks documentation to use fitgeotrans.

Table. 1. Projective transformation matrix

|  |  |  |
| --- | --- | --- |
| **3x3 Transformation Matrix** | | |
| 1.192233 | -0.2642 | 0.000233 |
| 1.264489 | 2.436467 | 0.005419 |
| -205.895 | -30.4412 | 1 |

The matrix is then applied to the image using the imwarp\* function {267}. Lastly, the book is cropped from the image for viewing purposes, using the image dimentions found previously after transformation {264-270}. From the “Transformed Image” in fig. 8, the bottom right of the book is clear, and gradually gets blurer towards the top left. This may be due to the resolution of the top left of the book being lower than the resolution of the bottom right of the book in the “Original” image.

\*The imwarp function and the imtranform function are interchangable, but it is recommended by the mathworks documentation to use imwarp.

Fig. 8. Result of Undoing Perspective Distortion of Planar Surface on image, “book.jpg”.