ANALYSIS OF CONTRAST ENHANCEMENT
METHODS FOR INFRARED IMAGES

A Thesis Presented to the Faculty of California Polytechnic State University,
San Luis Obispo

In Partial Fulfillment of the Requirements for the Degree
Master of Science in Electrical Engineering

By
Sprinkle Christian
December 2011
COMMITTEE MEMBERSHIP

TITLE: Analysis of Contrast Enhancement Methods for Infrared Images

AUTHOR: Sprinkle Christian

DATE SUBMITTED: December 16, 2011

COMMITTEE CHAIR: Dr. Wayne Pilkington
Associate Professor of Electrical Engineering

COMMITTEE MEMBER: Dr. William Ahlgren
Associate Professor of Electrical Engineering

COMMITTEE MEMBER: Dr. Fred DePiero
Professor of Electrical Engineering,
Associate Dean, College of Engineering
ABSTRACT

Analysis of Contrast Enhancement Methods for Infrared Images

Sprinkle Christian

Contrast enhancement techniques are used widely to improve the visual quality of images. The difference in luminance reflected from two adjacent surfaces creates contrast between the surfaces in the image. The greater the contrast, the easier it is to recognize and differentiate objects in an image. Thus object contrast is an important factor in the perception of the visual quality of an image and in its usefulness for object recognition and image analysis applications.

The focus of this thesis is on studying how well different contrast enhancement techniques developed for visible spectrum photographic images work on infrared images; to determine which techniques might be best suited for incorporation into commercial infrared imaging applications. The database for this thesis consists of night vision infrared images taken using a Photon 320 camera by FLIR Systems, Inc.

Numerous contrast enhancement techniques found in literature were reviewed in this project, out of which four (4) representative techniques have been selected and presented in detail. Homomorphic filtering, fuzzy-logic enhancement, and single-scale retinex techniques have been implemented based on published papers. These were each compared to the classical technique of histogram equalization using metrics of computational time, histogram standard deviation, sharpness and user observations. These metrics provide both quantitative and qualitative analyses of the implemented techniques which are relevant to the end user applications of infrared imaging.
Based on the metric calculations and results, homomorphic filtering and histogram equalization proved to have better results compared to the other techniques. All the implemented techniques are global contrast enhancement methods and for future work local contrast enhancement techniques may be applied to the results obtained in this investigation as a post-processing technique.
ACKNOWLEDGEMENTS

I dedicate this thesis to my family, whose guidance, support and positive attitude have brought me towards the completion of my academic career at Cal Poly.

I would also like to thank Daniel Kohler, whose motivation and feedback helped me see this thesis through.

I would take this opportunity to express my gratitude towards Dr. Wayne Pilkington who guided my work, gave me positive feedback and gave me freedom in direction with this project. I would like to thank him for his time commitment towards guiding me and reviewing my work. I would also like to thank Dr. Fred DePiero and Dr. William Ahlgren for serving on my thesis committee and for reviewing my work and giving me positive feedback.

The Photon 320 IR camera used in this research was provided by FLIR Systems, Inc. to the Cal Poly Electrical Engineering Department for research and development purposes. I would like to thank FLIR Systems, Inc. for providing me this opportunity to learn about the IR camera and its operation.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>ix</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>12</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>12</td>
</tr>
<tr>
<td>1.2 Previous Work in this Area</td>
<td>14</td>
</tr>
<tr>
<td>1.3 Scope</td>
<td>15</td>
</tr>
<tr>
<td>1.4 Organization of Thesis</td>
<td>16</td>
</tr>
<tr>
<td>2. PHOTON 320 CAMERA AND IMAGE ACQUISITION</td>
<td>17</td>
</tr>
<tr>
<td>2.1 Photon 320 Specifications [1]</td>
<td>18</td>
</tr>
<tr>
<td>2.2 Photon 320 System Configuration [1]</td>
<td>19</td>
</tr>
<tr>
<td>2.3 Hardware Setup</td>
<td>20</td>
</tr>
<tr>
<td>2.4 Software</td>
<td>22</td>
</tr>
<tr>
<td>3. HISTOGRAM EQUALIZATION</td>
<td>23</td>
</tr>
<tr>
<td>4. HOMOMORPHIC FILTERING</td>
<td>30</td>
</tr>
<tr>
<td>5. FUZZY LOGIC</td>
<td>38</td>
</tr>
<tr>
<td>6. SINGLE SCALE RETINEX</td>
<td>47</td>
</tr>
<tr>
<td>7. METRICS AND RESULTS</td>
<td>53</td>
</tr>
<tr>
<td>7.1 User Observations</td>
<td>53</td>
</tr>
<tr>
<td>7.2 Computational Complexity</td>
<td>54</td>
</tr>
<tr>
<td>7.3 Image Sharpness</td>
<td>55</td>
</tr>
<tr>
<td>7.4 Histogram Standard Deviation</td>
<td>57</td>
</tr>
<tr>
<td>7.5 Image Results</td>
<td>60</td>
</tr>
<tr>
<td>8. CONCLUSION AND FUTURE WORK</td>
<td>81</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>82</td>
</tr>
<tr>
<td>APPENDIX A</td>
<td>85</td>
</tr>
<tr>
<td>Software Configuration Procedure</td>
<td>85</td>
</tr>
<tr>
<td>Image Capture Procedure</td>
<td>91</td>
</tr>
<tr>
<td>APPENDIX B [21]</td>
<td>93</td>
</tr>
<tr>
<td>APPENDIX C</td>
<td>94</td>
</tr>
<tr>
<td>APPENDIX D [22]</td>
<td>97</td>
</tr>
<tr>
<td>APPENDIX E</td>
<td>98</td>
</tr>
<tr>
<td>APPENDIX F</td>
<td>99</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 1 Photon 320 Specifications ................................................................. 18
Table 2 Photon 320 Accessories Used For This Project [1] ......................... 20
Table 3 Photon 320 Hardware Manufacturer and Model Number .................. 21
Table 4 Computational Complexity of Test Images ..................................... 55
Table 5 Sharpness of Test Images ................................................................. 57
Table 6 Standard Deviation of Test Images .................................................. 58
LIST OF FIGURES

Figure 1 Photon 320 Camera .................................................................................................................. 17
Figure 2 Photon 320 Spectral Response .............................................................................................. 19
Figure 3 Photon 320 setup using Photon accessory kit ......................................................................... 19
Figure 4 Block Diagram of Camera Setup .............................................................................................. 20
Figure 5 Photo of Camera Setup ........................................................................................................... 20
Figure 6 Results of Histogram Equalization: (a) Original Image, (b) Histogram of (a), (c) Histogram equalization image, (d) Histogram of (c), (e) Scaled histogram equalization image, (f) Histogram of (e) ................................................................................................. 25
Figure 7 Results of Histogram Equalization: (a) Original Image, (b) Histogram of (a), (c) Histogram equalization image, (d) Histogram of (c), (e) Scaled histogram equalization image, (f) Histogram of (e) ................................................................................................. 26
Figure 8 Results of Histogram Equalization: (a) Original Image, (b) Histogram of (a), (c) Histogram equalization image, (d) Histogram of (c), (e) Scaled histogram equalization image, (f) Histogram of (e) ................................................................................................. 27
Figure 9 Results of Histogram Equalization: (a) Original Image, (b) Histogram of (a), (c) Histogram equalization image, (d) Histogram of (c), (e) Scaled histogram equalization image, (f) Histogram of (e) ................................................................................................. 28
Figure 10 Homomorphic Filtering Procedure [13] .............................................................................. 31
Figure 11 Algorithm for implementing homomorphic filtering ............................................................. 34
Figure 12 Results of Homomorphic Filtering: (a) Original Image, (b) Histogram of (a), (c) Homomorphic filtered image, (d) Histogram of (c), (e) Scaled homomorphic filtered image, (f) Histogram of (e) ................................................................................................. 34
Figure 13 Results of Homomorphic Filtering: (a) Original Image, (b) Histogram of (a), (c) Homomorphic filtered image, (d) Histogram of (c), (e) Scaled homomorphic filtered image, (f) Histogram of (e) ................................................................................................. 35
Figure 14 Results of Homomorphic Filtering: (a) Original Image, (b) Histogram of (a), (c) Homomorphic filtered image, (d) Histogram of (c), (e) Scaled homomorphic filtered image, (f) Histogram of (e) ................................................................................................. 36
Figure 15 Results of Homomorphic Filtering: (a) Original Image, (b) Histogram of (a), (c) Homomorphic filtered image, (d) Histogram of (c), (e) Scaled homomorphic filtered image, (f) Histogram of (e) ................................................................................................. 37
Figure 16 Input Membership Function for Fuzzy Rule-Based Contrast Enhancement .......................... 40
Figure 17 Output Membership Function for Fuzzy Rule-Based Contrast Enhancement ...................... 41
Figure 18 Flow chart for the implemented fuzzy logic process ............................................................. 42
Figure 19 Results of Fuzzy Logic: (a) Original Image, (b) Histogram of (a), (c) Fuzzy logic image, (d) Histogram of (c), (e) Scaled fuzzy logic image, (f) Histogram of (e) ................................................................................................. 43
Figure 20 Results of Fuzzy Logic: (a) Original Image, (b) Histogram of (a), (c) Fuzzy logic image, (d) Histogram of (c), (e) Scaled fuzzy logic image, (f) Histogram of (e) ................................................................................................. 44
Figure 21 Results of Fuzzy Logic: (a) Original Image, (b) Histogram of (a), (c) Fuzzy logic image, (d) Histogram of (c), (e) Scaled fuzzy logic image, (f) Histogram of (e) .................................................. 45

Figure 22 Results of Fuzzy Logic: (a) Original Image, (b) Histogram of (a), (c) Fuzzy logic image, (d) Histogram of (c), (e) Scaled fuzzy logic image, (f) Histogram of (e) .................................................. 46

Figure 23 Results of Single Scale Retinex: (a) Original Image, (b) Histogram of (a), (c) Single scale retinex image, (d) Histogram of (c), (e) Scaled single scale retinex image, (f) Histogram of (e) .................................................. 49

Figure 24 Results of Single Scale Retinex: (a) Original Image, (b) Histogram of (a), (c) Single scale retinex image, (d) Histogram of (c), (e) Scaled single scale retinex image, (f) Histogram of (e) .................................................. 50

Figure 25 Results of Single Scale Retinex: (a) Original Image, (b) Histogram of (a), (c) Single scale retinex image, (d) Histogram of (c), (e) Scaled single scale retinex image, (f) Histogram of (e) .................................................. 51

Figure 26 Results of Single Scale Retinex: (a) Original Image, (b) Histogram of (a), (c) Single scale retinex image, (d) Histogram of (c), (e) Scaled single scale retinex image, (f) Histogram of (e) .................................................. 52

Figure 27 Prewitt kernel for horizontal edge detection .................................................. 56

Figure 28 Prewitt kernel for vertical edge detection .................................................. 56

Figure 29 Metric Masks: (a) Original test image – high contrast (b) Image with mask extracted from (a) (c) Mask used for metric calculation (d) Original test image – normal contrast (e) Image with mask extracted from (d) (f) Mask used for metric calculation (g) Original test image – details in shadow (h) Image with mask extracted from (g) (i) Mask used for metric calculation .................................................. 59

Figure 30 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e) .................................................. 62

Figure 31 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e) .................................................. 64

Figure 32 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e) .................................................. 66

Figure 33 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g)
(k) Scaled single scale retinex applied (l) Histogram of (e) .............................. 68

Figure 34 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e) .............................. 69

Figure 35 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e) .............................. 71

Figure 36 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e) .............................. 73

Figure 37 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e) .............................. 75

Figure 38 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e) .............................. 76

Figure 39 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e) .............................. 78

Figure 40 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e) .............................. 80
1. INTRODUCTION

1.1 Introduction

Extending human vision with electronic imaging is a powerful and increasingly pervasive technique used in many commercial, military, law-enforcement, rescue, and healthcare applications.[1] Enabling people to see light in the parts of the electromagnetic spectrum outside the range of human visual sensing opens new vistas and possibilities for detecting or discerning sometimes life-critical information. For example, medical x-rays using radiation at very short wavelengths enable physicians to detect broken bones or cancerous growths inside the human body. Conversely infrared, or IR imaging systems, detect longer wavelength light to enable people to “see in the dark” (night vision) and to locate heat sources without external illumination.

Infrared literally means "below red," as infrared light has less energy and lower wavelengths than visible red light (the lowest wavelength color detectable by the human eye). The infrared portion of the electromagnetic spectrum ranges in wavelength from 1 to 15 microns, or about 2 to 30 times longer wavelength (and 2-30 times less energy) than visible light.

Infrared light is invisible to the unaided eye, but can be felt as heat by the temperature receptors in the skin. Warm objects emit infrared energy, and the hotter the object, the shorter the wavelength of IR energy emitted. This IR emission enables rescue workers equipped with long-wave IR sensors to locate a lost person in a deep forest in total darkness, for example. Infrared light can penetrate smoke and fog better than visible light, revealing objects that are normally obscured. It can also be used to detect the presence of excess heat or cold in a piece of machinery or a chemical reaction. Infrared
Camera technology is advancing rapidly, with increased performance in smaller packages at lower cost, enabling the use of IR cameras in many new applications. [1] Below is a list of some of the industrial, home, military and civilian applications of IR cameras:

- **Military**: Weapon siting, surveillance systems, night vision for drivers and pilots
- **Civilian Security**: Surveillance systems
- **Automotive**: Aid driver with night vision [1]
- **Firefighters**: Detect survivors in smoky rooms; forest fire detection
- **Commercial**: Photovoltaic cell, module and system testing
- **Energy**: Auditing houses for heater/cooler systems to detect if there are any air leaks from the doors and/or windows

Contrast enhancement of infrared (IR) images to make them more effective in such applications is the principle goal of this investigation. Contrast is created in an image by the difference in luminance reflected from two adjacent surfaces [3]. The greater the contrast, the easier it is to recognize and differentiate objects in an image. Thus object contrast is an important factor in the perception of the visual quality of an image and in its usefulness for object recognition and image analysis applications.

Contrast enhancement is one of the most common methods of low-level digital image processing. The field of *digital image processing* refers to processing digital images by means of a digital computer. A low-level process is characterized by the fact that both its inputs and outputs are images. Low-level processes involve primitive operations such as noise reduction, contrast enhancement and image sharpening [1]. The goal of contrast enhancement is to provide a more appealing image, with easier differentiation of objects, and improved clarity of object features and surface details.
Contrast enhancement finds application in many imaging fields, including infrared image processing [1].

1.2 Previous Work in this Area

A considerable amount of research has been done in the field of contrast enhancement. The field of contrast enhancement can be broadly divided into two major areas of research: (1) Global contrast enhancement techniques and (2) Local contrast enhancement techniques. Global contrast enhancement techniques are simple and powerful but cannot adapt to the local brightness features of the input image because these techniques use only global information collected over the entire image. Local contrast enhancement techniques adjust picture element (pixel) values are over smaller regions of an image to improve the visualization of structures in both the darkest and the lightest portions of the image at the same time.

Global contrast enhancement techniques include histogram equalization, fuzzy logic enhancement, homomorphic filtering, and retinex theory. Much of the research toward local contrast enhancement techniques focused on improvement of histogram equalization based contrast enhancement. Such methods as mean preserving bi-histogram equalization (BBHE), equal area dualistic sub-image histogram equalization (DSIHE), and minimum mean brightness error bi-histogram equalization (MMBEBHE) [4] seek to provide the benefits of localize processing to the basic principles of the histogram equalization method. BBHE separates the input image histogram into two parts based on input mean. After separation, each part is equalized independently. This method tries to overcome the brightness preservation problem of basic histogram equalization.
1.3 Scope

The aim of this investigation is to find contrast enhancement techniques best suited for IR images using the Photon 320 camera manufactured by FLIR Systems, Inc. All the images have been captured using the Photon 320 camera. The test image database consists of images taken in a room during the night with either no light or low light conditions, and images taken in a room during daylight on a hazy day. Four global contrast enhancement techniques have been analyzed namely: histogram equalization, homomorphic filtering, fuzzy logic enhancement, and single-scale retinex. The techniques have been compared with metrics including computational complexity, sharpness, standard deviation and user observations.

For the purpose of this investigation, only global contrast enhancement techniques have been chosen for study, implementation and comparison. To enhance the quality of results, local contrast enhancement techniques may be applied for post processing of the images if needed in the end-user application.

For the purpose of this investigation, the pixel values of all the output images from each enhancement technique are rescaled to maximize the use of the dynamic range of display. Thus, the image pixel values have been scaled and offset so that the minimum pixel value is always 0, and the maximum pixel value is 255 in the image. This step, also known as histogram expansion, is appropriate in this investigation because it mimics the operation of an Automatic Gain Control (AGC) amplifier in the camera or subsequent display preprocessing.
1.4 Organization of Thesis

This thesis integrates the results from four different approaches of contrast enhancement techniques. Chapter 2 deals with the Photon 320 camera and its hardware setup (The software setup is discussed in Appendix A). Chapter 3 discusses the Histogram Equalization method using the Matlab built-in function `histeq`. Chapters 4, 5 and 6 cover contrast enhancement by homomorphic filtering, fuzzy logic and single-scale retinex respectively. Chapter 7 provides a comparison of all the four techniques using metrics like computational complexity, sharpness, standard deviation, and user observations.
2. PHOTON 320 CAMERA AND IMAGE ACQUISITION

Photon 320 [1] *

The Photon 320 is a commercial market, military qualified (CDMQ), 320 x 240 pixel long-wave infrared thermal imaging camera core manufactured by FLIR Systems, Inc. It provides a low cost IR imaging solution for applications needing a wide range of operating temperatures, using an uncooled microbolometer sensor array. The Photon 320 camera generates 8- or 14-bit resolution digital video at a 30 Hz frame rate. Over 40,000 Photon camera cores are currently in use, in a variety of applications including handheld firefighting safety instruments and as a payload camera for small unmanned vehicles.

The Photon camera line was recently discontinued by FLIR, and replaced with an enhanced model: the Tau series cameras.

![Photon 320 Front View](image1)

![Photon 320 Rear View](image2)

*Figure 1 Photon 320 Camera*
### 2.1 Photon 320 Specifications [1] *

*Table 1 Photon 320 Specifications*

<table>
<thead>
<tr>
<th><strong>System Overview</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System Type</strong></td>
<td>Uncooled LWIR Thermal Imager</td>
</tr>
<tr>
<td><strong>Sensor Type</strong></td>
<td>VOx Microbolometer</td>
</tr>
<tr>
<td><strong>Sensor Format</strong></td>
<td>324 x 256</td>
</tr>
<tr>
<td><strong>Pixel Size</strong></td>
<td>38(\mu m)</td>
</tr>
<tr>
<td><strong>Spectral Band</strong></td>
<td>7.5 - 13.5 (\mu m)</td>
</tr>
<tr>
<td><strong>NEdT Performance</strong></td>
<td>&lt;85mK at f/1.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Outputs</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Analog Video</strong></td>
<td>NTSC @ 30 Hz</td>
</tr>
<tr>
<td></td>
<td>Option: NTSC slow (7.5 Hz)</td>
</tr>
<tr>
<td></td>
<td>Option: PAL @ 25 Hz</td>
</tr>
<tr>
<td></td>
<td>Option: PAL slow (8.3 Hz)</td>
</tr>
<tr>
<td><strong>Digital Video</strong></td>
<td>8- or 14-bit serial LVDS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Operation</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image Control</strong></td>
<td>Invert, revert, 2x digital zoom, digital detail enhancement (DDE)</td>
</tr>
<tr>
<td><strong>Camera Control</strong></td>
<td>Autonomous, or manual via GUI or RS-232 serial command</td>
</tr>
<tr>
<td><strong>Signal Interface</strong></td>
<td>30-pin SAMTEC connector for power, video, communication, digital data, external sync. Optional 15-pin D-Sub connector with Accessory Kit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Power</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input Voltage Range</strong></td>
<td>5 - 24 VDC</td>
</tr>
<tr>
<td><strong>Power Dissipation</strong></td>
<td>1.6 W steady-state</td>
</tr>
<tr>
<td><strong>Time to Image</strong></td>
<td>~2 seconds</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Environmental</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operating Temp Range</strong></td>
<td>-40 - +80 C external temp</td>
</tr>
<tr>
<td><strong>Scene Temp Range</strong></td>
<td>To 150 C standard</td>
</tr>
</tbody>
</table>
Shock
To 70 g’s, 11ms half-sine profile

<table>
<thead>
<tr>
<th>Physical Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
</tr>
<tr>
<td>Weight</td>
</tr>
<tr>
<td>Mounting Interface</td>
</tr>
</tbody>
</table>

Spectral Sensitivity [1] *

The Photon 320 system uses a long-wave infrared sensor, with a wavelength sensitivity profile shown below:

![Figure 2 Photon 320 Spectral Response](image)

2.2 Photon 320 System Configuration [1] *

The Photon 320 system shown below, using the Photon accessory kit to interface the camera to a PC, was used for the purposes of this research:

![Figure 3 Photon 320 setup using Photon accessory kit](image)
2.3 Hardware Setup

Figure 4 Block Diagram of Camera Setup

Figure 5 Photo of Camera Setup

The table below describes the components used from the Photon 320 accessory kit for the purposes of this project:

Table 2 Photon 320 Accessories Used For This Project [1]*

<table>
<thead>
<tr>
<th>Accessory item</th>
<th>Functional Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6” Interface Cable, Photon to I/O Module</td>
<td>Provides electrical interface by means of a 15-pin mating connector at the camera end, and 18-pin connector at the I/O module end.</td>
</tr>
</tbody>
</table>
### AC/DC Power Supply & Line Cord
Provides nominal power [9 VDC] to operate the Photon

### Ethernet Module
This 100/1000 baseT Ethernet Interface module allows for camera control along with real-time streaming uncompressed video data from the Photon via standard Ethernet hardware. The adapter auto senses network capability and runs at standard 100-megabit or full gigabit Ethernet speed. The module includes the Ethernet interface adapter and camera cabling. The module allows capture of both 8-bit data and the full 14-bit bandwidth digital video. Analog video is also output via a BNC connector.

### Hardware

*Table 3 Photon 320 Hardware Manufacturer and Model Number*

<table>
<thead>
<tr>
<th>Nomenclature</th>
<th>Manufacturer and Model Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet Module / Frame Grabber</td>
<td>iPORT PT1000-IDG-LVDS-V2-V434</td>
</tr>
<tr>
<td>IR Camera</td>
<td>Photon 320</td>
</tr>
<tr>
<td>RS-422 cable</td>
<td>FLIR 308-0091-00 REV. 110 RND 43/08</td>
</tr>
<tr>
<td>Ethernet cable</td>
<td>(Generic)</td>
</tr>
</tbody>
</table>
2.4 Software

- Coyote (Indigo Systems Inc.), version 2.2.1 build 441

- Photon GUI (FLIR), version 2.4.0
3. HISTOGRAM EQUALIZATION

Histogram equalization (HE) is a commonly used global contrast enhancement technique for both color and grayscale images. HE spreads out and flattens the histogram of the number of image pixels at each gray level value, thus stretching the intensity values in the image over more of the available dynamic range of gray-levels and increasing the apparent contrast in the image. This method is especially useful when an image is represented by close contrast values, such as images in which both the background and foreground are both bright, or else both are dark at the same time.

Histogram Equalization is accomplished by linearizing the cumulative density function of the image intensity levels. Consider a discrete grayscale image and let \( n_i \) be the number of occurrences of gray level \( i \). A normalized histogram of the image shows the probability of occurrence of a pixel of level \( i \) in the image, and would be given by a collection of probability values for each pixel level:

\[
p(x_i) = \text{probability that pixel } x \text{ has gray level } i = \frac{n_i}{n}
\]

where \( n = \) the total number of pixels in the image. The cumulative density function for this histogram would be given by:

\[
cdf_x(i) = \sum_{j=0}^{i} p(x_j)
\]

Histogram equalization seeks a transformation of input pixel values to output pixel values that will make this cumulative density function as nearly linear as possible across the range of pixel gray levels. That is

\[
cdf_x(i) = K \cdot i
\]
The algorithm for implementing classical histogram equalization is shown below from [5]:

```plaintext
for each pixel p in the image do
    deposit p in temporary bin bg(p).
end for

j ← 0
for each temporary bin bi do
    Increment count
    Copy pixels bi into histogram bin h(j/D).
    j ← j + |bi| \{|bi| = number of pixels in bi\}
end for

for i = 0 to D − 1 do
    Set grey level of each pixel in bin hi to i.
end for
```

While easy to implement and not very computationally intensive, HE has several known drawbacks. First, HE does not preserve the average brightness of the input image in the output image. Therefore, the processed output image will often appear unnaturally bright or “washed out.” Also, the method is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal. To overcome these drawbacks, numerous variations of the classic HE technique have been published [6].

For the purposes of this project, global histogram equalization is chosen as the “gold standard” for comparing alternative methods of contrast enhancement. The Matlab function `histeq` was used as the particular reference method for contrast enhancement implementation. The `histeq` function enhances the contrast of images by transforming the values in an intensity image, or the values in the color map of an indexed image [7].

The following images demonstrate the results of histogram equalization on infrared image examples.
Figure 6  Results of Histogram Equalization: (a) Original Image, (b) Histogram of (a), (c) Histogram equalization image, (d) Histogram of (c), (e) Scaled histogram equalization image, (f) Histogram of (e)
Figure 7 Results of Histogram Equalization: (a) Original Image, (b) Histogram of (a), (c) Histogram equalization image, (d) Histogram of (c), (e) Scaled histogram equalization image, (f) Histogram of (e)
Figure 8  Results of Histogram Equalization: (a) Original Image, (b) Histogram of (a), (c) Histogram equalization image, (d) Histogram of (c), (e) Scaled histogram equalization image, (f) Histogram of (e)
Figure 9  Results of Histogram Equalization:  (a) Original Image, (b) Histogram of (a), (c) Histogram equalization image, (d) Histogram of (c), (e) Scaled histogram equalization image, (f) Histogram of (e)
Notice that for images with single-mode, normal histogram distributions, such as those in figures 7 through 9, histogram expansion by pixel value scaling to utilize the full dynamic range shown in parts (e) and (f) is not necessary, as histogram equalization by itself spreads the pixel values over nearly the full range. For images with disproportionately large histogram values for the darkest or lightest pixel values, such as the image in Figure 6, histogram equalization will still benefit from rescaling the equalized levels to use the full dynamic range.
4. HOMOMORPHIC FILTERING

Homomorphic filtering is a frequency domain method for contrast enhancement. It has been used in a variety of applications like shadow identification [8], underwater image pre-processing [9], [10], contrast enhancement for raised or indented characters [11], and seismic data processing [12]. Homomorphic filtering sharpens features in an image by enhancing high frequencies and sharpening object edges [11]. It also flattens lighting variations in an image, bringing details out of shadows. It provides simultaneous dynamic range compression (reducing illumination variation) and contrast enhancement (increasing reflectance variation). Homomorphic filtering can thus prove to be most effective on images that have large variations in lighting.

Homomorphic filtering is based on a simple model of the imaging process, wherein images are formed from sensing light from an illumination source reflected from the surface of the objects being observed. An image, characterized as a two-dimensional intensity pattern $f(x,y)$, can then be decomposed into two different components:

$$f(x,y) = i(x,y) \cdot r(x,y)$$

1. The amount and spatial distribution of source illumination incident on the object $i(x,y)$, and
2. The two-dimensional reflectivity profile of the object, $r(x,y)$

These are often referred to as the \textit{illumination} and the \textit{reflectance} components respectively of an image. These component spatial distributions are combined by simple scalar multiplication to form the image brightness pattern incident on the camera, as shown in

$$f(x,y) = i(x,y) \cdot r(x,y)$$

(1)
The illumination distribution arriving at the reflecting objects in an image results from the lighting conditions present when the image is captured, and will change when any lighting source changes position, intensity, or illumination pattern [13]. The illumination component is assumed to be characterized by slow spatial variations (low spatial frequencies). The object reflectance profile results from the way the objects in the image reflect light, and is determined by the intrinsic properties of the object itself – the surface shape, roughness, material reflectivity, etc. [13]. The reflectance component is assumed to be dominated by abrupt spatial variations (high spatial frequencies).

The model of image formation as shown in \( f(x,y) = i(x,y) \cdot r(x,y) \) (1) is well known as the illumination-reflectance model and can be used to address the problem of improving the quality of an image that has been acquired under poor illumination conditions [13]. Figure 1 shows the implementation procedure for homomorphic filtering.

![Figure 10 Homomorphic Filtering Procedure [13]](image)

To apply homomorphic filtering, the illumination and the reflectance components from \( f(x,y) = i(x,y) \cdot r(x,y) \) (1) must be separated in the spatial frequency domain so that they can be treated independently. Fourier transformation from the spatial domain to the spatial frequency domain, however, is linear and associative only under addition; not under multiplication,

\[
\text{FFT}\{f(x,y)\} \neq \text{FFT}\{i(x,y)\} \cdot \text{FFT}\{r(x,y)\}
\]  
(2)
Rather,

\[ \text{FFT}\{f(x,y)\} = \text{FFT}\{i(x,y)\} \ast \text{FFT}\{r(x,y)\} \]  (3)

where the Fourier Transforms of the illumination and reflectances are convolved rather than multiplied in the spatial frequency domain.

Thus, a logarithm operation is performed on the image distribution in (1) before transformation to the frequency domain, which allows separation of the illumination and reflectance components.

\[ f(x,y) = \ln[f(x,y)] = \ln[i(x,y) \cdot r(x,y)] \]  (4)

\[ f(x,y) = \ln[i(x,y)] + \ln[r(x,y)] \]  (5)

\[ f(x,y) = i'(x,y) + r'(x,y) \]  (6)

Taking the two-dimensional FFT of both sides:

\[ F(a,b) = I'(u,v) + R'(u,v) \]  (7)

where,

\[ u, v = \text{spatial frequencies in the x, y directions, respectively;} \]

\[ I'(u,v) = 2\text{-D Fourier Transform of } \ln[i(x,y)] \]

\[ R'(u,v) = 2\text{-D Fourier Transform of } \ln[r(x,y)]. \]

Now applying an application specific filter frequency response function \( H(a,b) \) to the image spatial frequency distribution yields a filtered image spectrum:

\[ G(u,v) = F(u,v) H(u,v) = I'(u,v) \cdot H(u,v) + R'(u,v) \cdot H(u,v) \]  (8)

At low spatial frequencies, in the range of frequencies where the illumination function has most of its energy, the filter function \( H(u,v) \) has a value of \( \mu_L << 1 \). At higher spatial
frequencies, where the details in the reflectance function reside, the filter function $H(u,v)$ has a value of $\mu_H > 1$. Therefore, the filtered image spectrum can be approximated by:

$$G(u,v) \approx I'(u,v) \mu_L + R'(u,v) \mu_H$$  \hspace{1cm} \text{(9)}$$

Taking the inverse FFT yields the filtered image function (spatial distribution)

$$g(x,y) \approx \text{IFFT}[G(u,v)] = \mu_L \ln[i(x,y)] + \mu_H \ln[r(x,y)]$$

$$g(x,y) \approx \mu_H \ln[r(x,y)], \quad \text{for } \mu_L << 1$$  \hspace{1cm} \text{(10)}$$

Figure 11 *Algorithm for implementing homomorphic filtering* below depicts the algorithm used to implement the homomorphic filter used in this project. The algorithm is based on the equations above.
Figure 11 Algorithm for implementing homomorphic filtering

The following are the image outputs after homomorphic filtering was applied on the original images:

Figure 12 Results of Homomorphic Filtering: (a) Original Image, (b) Histogram of (a), (c) Homomorphic filtered image, (d) Histogram of (c), (e) Scaled homomorphic filtered image, (f) Histogram of (e)
Figure 13  Results of Homomorphic Filtering: (a) Original Image, (b) Histogram of (a), (c) Homomorphic filtered image, (d) Histogram of (c), (e) Scaled homomorphic filtered image, (f) Histogram of (e)
Figure 14 Results of Homomorphic Filtering: (a) Original Image, (b) Histogram of (a), (c) Homomorphic filtered image, (d) Histogram of (c), (e) Scaled homomorphic filtered image, (f) Histogram of (e)
Figure 15 Results of Homomorphic Filtering: (a) Original Image, (b) Histogram of (a), (c) Homomorphic filtered image, (d) Histogram of (c), (e) Scaled homomorphic filtered image, (f) Histogram of (e)
5. FUZZY LOGIC

Fuzzy logic has been successfully applied to image enhancement and classification for many years [14], [15]. The foundation of fuzzy set theory was first established in 1965 by Lotfi Zadeh. The theory of fuzzy sets is a theory of graded concepts, a theory in which everything is a matter of degree [14]. Unlike two-valued Boolean logic, fuzzy logic is based on degrees of membership and degrees of truth. Fuzzy logic not only recognizes true and false values but is also useful for propositions that can be represented with varying degrees of truth and falseness.

The fuzzy design process, as explained by [16] proceeds with the following steps:

1. The process begins with an inventory of the sensor inputs available to the process.
2. The anticipated range of each of the inputs should be known, and is divided into several coarse, overlapping ‘Membership Functions,’ also known as ‘Classes’ or ‘Fuzzy Sets’.
3. A similar set of ‘Output Membership Functions’ is defined covering the range of each output.
4. Every possible combination of input membership classes is related to an output class via a set of logic-like sentences. These rules take the general form of:
   IF (some conjunction of input IS input member function), THEN (output IS output member function)

   The collection of all these sentences forms a set of ‘Fuzzy Rules’ in a ‘Rule Base’.

   The process of generating this rule base is called ‘Implication’.
5. An appropriate method for establishing the degree of truth of an output membership function based on the relative degree of truth of each of the input
functions mapped to it by the fuzzy rules is decided. This is called the ‘Conjunction Method’.

6. An appropriate method is further decided for mapping output membership functions, each with a certain degree of truth, back to the input variables. This process is called ‘Defuzzification’.

7. If there are multiple independent fuzzy processes, the outputs of each can be optionally combined to generate the ultimate outputs of the fuzzy system. The function that may be selected to combine these is called the ‘Aggregation’ method.

One possible approach to infrared image contrast enhancement using fuzzy logic is by means of a Takagi-Sugeno fuzzy rule based system [1]. Takagi-Sugeno rules have consequents (THEN outcomes) that are numeric functions (generally linear combinations) of the input values. This approach, then, to enhancing the contrast of a gray-scale image proposes the following rules:

- IF a pixel is dark, THEN make it darker
- IF a pixel is gray, THEN make it mid-gray
- IF a pixel is bright, THEN make it brighter

Membership functions characterize the fuzziness in a fuzzy set and map all the elements of a set into real numbers in the range [0, 1] [15]. When the value of membership is higher, the “truth” that the set element belongs to that particular member function is higher too.

The input membership functions for an image contrast enhancement system are shown below in Figure 1. Here, the set of all possible input image pixel values is mapped
to three (3) linguistic terms: Dark, Gray and Bright. The values $\mu_i(z)$ quantify the “degree of membership” of a particular input pixel intensity value to the each of the three member functions (Dark, Gray, or Bright; denoted by the subscript $i$). Thus, $\mu_{\text{dark}}(z)$ assigns a value between 0 and 1 to how truly “dark” a given input pixel intensity value $z$ is. Similarly, $\mu_{\text{gray}}(z)$ and $\mu_{\text{bright}}(z)$ characterize how truly Gray or Bright a pixel value $z$ is. The Dark and Bright input membership functions have been implemented by using sigmoid functions, whereas the Gray input membership function has been implemented by a Gaussian function. The sigmoid function, also known as the logistic function, is an ‘S’ shaped function that is continuous and non-linear; and is defined mathematically as follows:

$$f(x) = \frac{1}{1 + e^{-g(x)}}$$

(11)

where $x$ is the input and $g(x)$ is the gain. The Gaussian function is defined as below:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma^2} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

(12)

The input membership function for fuzzy rule-based contrast enhancement is shown in Figure 16 _Input Membership Function for Fuzzy Rule-Based Contrast Enhancement_.

![Input Membership Function for Fuzzy Rule-Based Contrast Enhancement](image-url)
Three linguistic terms are also defined here for the output member functions; and these are referred to as *Darker*, *Mid-gray* and *Brighter*. As is common in some implementations of Takagi-Sugeno systems, the output fuzzy sets are defined as fuzzy singletons – that is the output membership functions are single-valued constants. For the purposes of this project, the singleton output membership function values have been selected as follows:

\[
\begin{align*}
Darker &= 0 \ (ν_d) \\
Mid-gray &= 127 \ (ν_g) \\
Brighter &= 255 \ (ν_b)
\end{align*}
\]

These are shown below in Figure 17 *Output Membership Function for Fuzzy Rule-Based Contrast Enhancement*:

![Output Membership Function for Fuzzy Rule-Based Contrast Enhancement](image)

For a Takagi-Sugeno system design, the fuzzy logic rules that determine the outputs of the system will use the following linear combination of input and output member function values. Since the output membership functions are constants, the output \(ν_0\) to any input \(z_0\), is given by:

\[
ν_0 = \frac{[μ_{dark}(z_0) \times ν_d + μ_{gray}(z_0) \times ν_g + μ_{bright}(z_0) \times ν_b]}{[μ_{dark}(z_0) + μ_{gray}(z_0) + μ_{bright}(z_0)]}
\]

(13)
where,

\[ \mu_{\text{dark}}(z), \mu_{\text{gray}}(z) \text{ and } \mu_{\text{bright}}(z) = \text{the input pixel intensity values and} \]
\[ v_d, v_g \text{ and } v_b = \text{the output pixel intensity values} \]

This relationship accomplishes the processes of implication, aggregation and defuzzification together with a straightforward numeric computation.

For the applications of image processing, fuzzy logic is generally computationally intensive, since it requires performing fuzzification, processing of all the rules, implication, aggregation and defuzzification on every pixel in the input image. Using a Takagi-Sugeno design with singleton output membership functions reduces computational time significantly by simplifying the computational time requirements in implication, aggregation and defuzzification.

Figure 18 *Flow chart for the implemented fuzzy logic process* shows the block diagram of the process developed for the fuzzy logic technique implemented for this project.
Figure 19 Results of Fuzzy Logic: (a) Original Image, (b) Histogram of (a), (c) Fuzzy logic image, (d) Histogram of (c), (e) Scaled fuzzy logic image, (f) Histogram of (e)
Figure 20 Results of Fuzzy Logic: (a) Original Image, (b) Histogram of (a), (c) Fuzzy logic image, (d) Histogram of (c), (e) Scaled fuzzy logic image, (f) Histogram of (e)
Figure 21 Results of Fuzzy Logic: (a) Original Image, (b) Histogram of (a), (c) Fuzzy logic image, (d) Histogram of (c), (e) Scaled fuzzy logic image, (f) Histogram of (e)
Figure 22 Results of Fuzzy Logic: (a) Original Image, (b) Histogram of (a), (c) Fuzzy logic image, (d) Histogram of (c), (e) Scaled fuzzy logic image, (f) Histogram of (e)
6. SINGLE SCALE RETINEX

The idea of Retinex was proposed as a model of lightness and color perception of the human vision. The basic idea of Retinex algorithm is to separate illumination from the reflectance in a given image. It is an algorithm that improves the contrast, brightness and sharpness if an image. Different algorithms have been developed to implement the Retinex model and concept. Single-scale Retinex (SSR), Multiscale Retinex (MSR) and Multiscale Retinex with Color Restoration (MSRCR) have evolved since the idea of Retinex was first proposed [18], [19], [20]. The Retinex is an image enhancement algorithm that improves the brightness, contrast and sharpness of an image. The algorithm performs a non-linear spatial/spectral transform that provides simultaneous dynamic range compression and color constancy [18]. The Retinex is a member of the class of center surround functions where each output value of the function is determined by the corresponding input value (center) and its neighborhood (surround). For the retinex the center is defined as each pixel value and the surround is a Gaussian function [18]. The following equations define the mathematical form of single-scale retinex (SSR) [18]:

\[
R(x_1, x_2) = \alpha(\log(I(x_1, x_2)) - \log(I(x_1, x_2)) * F(x_1, x_2)) - \beta
\]  

(14)

where the ‘*’ represents convolution.

F is the Gaussian filter (surround or kernel) defined by:

\[
F(x_1, x_2) = K \exp\left[-\frac{x_1^2 + x_2^2}{\sigma^2}\right]
\]

(15)

Large Gaussian kernels, typically with \(\sigma\) ranging from 50 to 120, are normally used to produce good SSR performance, thus spatial domain convolution would be extremely time consuming.
The unique enhancement achieved by the retinex technique allows the use of the algorithm in applications such as medical radiography, underwater photography, and forensic investigations [18]. The retinex algorithm is computationally intensive and involves performing a large number of complex operations and data transfers.

\[
 f(x, y) * g(x, y) \Rightarrow F(\mu, \nu)G(\mu, \nu) \tag{16}
\]

\[
 R(x_1, x_2) = \alpha(\log(I(x_1, x_2)) - \log[F^{-1}(I'(\mu, \nu)F'((\mu, \nu)))] - \beta \tag{17}
\]
Figure 23 Results of Single Scale Retinex: (a) Original Image, (b) Histogram of (a), (c) Single scale retinex image, (d) Histogram of (c), (e) Scaled single scale retinex image, (f) Histogram of (e)
Figure 24 Results of Single Scale Retinex: (a) Original Image, (b) Histogram of (a), (c) Single scale retinex image, (d) Histogram of (c), (e) Scaled single scale retinex image, (f) Histogram of (e)
Figure 25 Results of Single Scale Retinex: (a) Original Image, (b) Histogram of (a), (c) Single scale retinex image, (d) Histogram of (c), (e) Scaled single scale retinex image, (f) Histogram of (e)
Figure 26 Results of Single Scale Retinex: (a) Original Image, (b) Histogram of (a), (c) Single scale retinex image, (d) Histogram of (c), (e) Scaled single scale retinex image, (f) Histogram of (e)
7. METRICS AND RESULTS

Test images have been chosen carefully to represent a wide variety of challenging contrast situations. The infrared images used here represent real imaging situations and challenges for IR image applications like night vision, security cameras, fire fighting and automobile night vision. A variety of images depicting low contrast, high contrast, features in shadows and features in highly saturated areas have been selected. The variety of the test images will prove to be useful while testing the effectiveness of the implemented methods.

The following metrics have been chosen for the purpose of this investigation:

1. User Observations
2. Computational Complexity
3. Image Sharpness and
4. Image Histogram Standard Deviation

7.1 User Observations

A set of users were chosen to carry out this investigation. The users represented a variety of occupations from software and electrical engineers to science and management. The users rated the images based on how well the objects in a particular image could be seen and the contrast and quality of the image. While making a decision the users were not informed about which image represented which technique of contrast enhancement. A set of 14 users gave their feedback and the rated the images on a scale of 1 to 5 where 1 represented the ‘best output’ according to the user.
Based on an average of user observation outputs, scaled homomorphic filtering was selected by users as providing the best results for contrast enhancement. Output images of scale homomorphic filtering were seen to have higher quality of contrast while not deleting information from the original test images but enhancing the details in the test images.

7.2 Computational Complexity

In this investigation, the Big O calculation is used to compute the computational complexity of the implemented algorithms. Big O notation is used in Computer Science to describe the performance or complexity of an algorithm. Big O specifically describes the worst-case scenario for the computational complexity of an algorithm.

Based on the histogram equalization algorithm explained in Chapter 3, the upper bound Big O calculation for histogram equalization technique is 3O (n). Similarly for the homomorphic filtering algorithm described in Appendix B, the upper bound Big O calculation for homomorphic filtering is O(n^2). From appendix C, the fuzzy logic upper bound Big O calculation is found to be O(n^2). The upper bound Big O calculation for single scale retinex is O(n).

Ideally, computational complexity can be calculated by analyzing the number of computations in the developed algorithm. Since histeq Matlab function is used as the gold standard, the technique could not be applied to calculate computational complexity. The current metric is simply a time measure of how long it takes the algorithm to run and generate a filtered output image. The computational time for the implemented techniques
may be significantly reduced by making changes in the algorithm to make it competitive with the Matlab \textit{histeq} inbuilt function.

The following table depicts the calculated computational complexity of the test images:

Table 4 \textit{Computational Complexity of Test Images}

<table>
<thead>
<tr>
<th>Original Image</th>
<th>Histogram Equalization (seconds)</th>
<th>Homomorphic Filtering (seconds)</th>
<th>Fuzzy Logic (seconds)</th>
<th>Single Scale Retinex (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 27</td>
<td>0.541</td>
<td>1.292</td>
<td>3.765</td>
<td>20.609</td>
</tr>
<tr>
<td>Figure 28</td>
<td>0.641</td>
<td>0.882</td>
<td>3.855</td>
<td>19.989</td>
</tr>
<tr>
<td>Figure 29</td>
<td>0.601</td>
<td>0.891</td>
<td>3.735</td>
<td>19.849</td>
</tr>
<tr>
<td>Figure 30</td>
<td>0.581</td>
<td>0.892</td>
<td>3.796</td>
<td>20.439</td>
</tr>
<tr>
<td>Figure 31</td>
<td>1.141</td>
<td>1.402</td>
<td>4.677</td>
<td>20.991</td>
</tr>
<tr>
<td>Figure 32</td>
<td>0.661</td>
<td>1.091</td>
<td>4.056</td>
<td>20.489</td>
</tr>
<tr>
<td>Figure 33</td>
<td>0.761</td>
<td>1.062</td>
<td>4.116</td>
<td>20.239</td>
</tr>
<tr>
<td>Figure 34</td>
<td>0.661</td>
<td>1.001</td>
<td>4.156</td>
<td>20.399</td>
</tr>
<tr>
<td>Figure 35</td>
<td>0.671</td>
<td>1.061</td>
<td>4.096</td>
<td>21.330</td>
</tr>
<tr>
<td>Figure 36</td>
<td>0.651</td>
<td>1.062</td>
<td>4.246</td>
<td>21.251</td>
</tr>
<tr>
<td>Figure 37</td>
<td>0.651</td>
<td>1.062</td>
<td>4.336</td>
<td>21.501</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>0.681</strong></td>
<td><strong>1.063</strong></td>
<td><strong>4.075</strong></td>
<td><strong>20.644</strong></td>
</tr>
</tbody>
</table>

The data in the table above shows the time taken by each algorithm to execute. This data can be further refined by customizing the Matlab code that has been developed for this investigation. Since the Matlab inbuilt function \textit{histeq} is used for histogram equalization, the implementation time in seconds for this technique is lower. Other techniques can be customized to lower their respective execution times.

7.3 Image Sharpness

The Prewitt operator from Matlab was used for detecting the sharpness of the output images compared with the input images. This is a discrete differentiation operator,
computing an approximation of the gradient of the image intensity function. The Prewitt operator value therefore shows how "abruptly" or "smoothly" the image changes at each point in an image. This metric helps in analyzing how blurry the output image gets while enhancing the contrast. Since contrast enhancement is used to enhance the overall visual clarity of the image, if in the process the output image becomes blurred then the contrast would not be prominent and effective. This metric helps understand the overall visual quality of the output image after the enhancement techniques have been applied. The higher the value of sharpness the better the contrast enhancement technique has preserved sharpness.

The Prewitt operator is based on convolving the image with a small, separable, and integer valued two-dimensional filter. The following kernel is used for horizontal and vertical edge detection for the Prewitt operator:

\[
\begin{array}{ccc}
-1 & -1 & -1 \\
0 & 0 & 0 \\
1 & 1 & 1 \\
\end{array}
\]

*Figure 27 Prewitt kernel for horizontal edge detection*

\[
\begin{array}{ccc}
-1 & 0 & 1 \\
-1 & 0 & 1 \\
-1 & 0 & 1 \\
\end{array}
\]

*Figure 28 Prewitt kernel for vertical edge detection*

The following table summarizes the calculated sharpness of the test images:
Table 5 Sharpness of Test Images

<table>
<thead>
<tr>
<th>Original Image</th>
<th>Original Image Sharpness</th>
<th>Scaled Original Image Sharpness</th>
<th>Scaled Histogram Equalization</th>
<th>Scaled Homomorphic Filtering</th>
<th>Scaled Fuzzy Logic</th>
<th>Scaled Single Scale Retinex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 27</td>
<td>10.54</td>
<td>21.08</td>
<td>20.90</td>
<td>21.17</td>
<td>21.60</td>
<td>21.22</td>
</tr>
<tr>
<td>Figure 28</td>
<td>15.73</td>
<td>31.45</td>
<td>40.46</td>
<td>31.62</td>
<td>40.57</td>
<td>31.75</td>
</tr>
<tr>
<td>Figure 29</td>
<td>10.39</td>
<td>20.78</td>
<td>37.41</td>
<td>24.88</td>
<td>30.90</td>
<td>24.87</td>
</tr>
<tr>
<td>Figure 30</td>
<td>7.71</td>
<td>20.79</td>
<td>50.34</td>
<td>19.48</td>
<td>24.24</td>
<td>19.52</td>
</tr>
<tr>
<td>Figure 31</td>
<td>12.74</td>
<td>25.48</td>
<td>31.02</td>
<td>23.60</td>
<td>26.78</td>
<td>25.65</td>
</tr>
<tr>
<td>Figure 32</td>
<td>11.71</td>
<td>23.42</td>
<td>30.61</td>
<td>20.23</td>
<td>20.30</td>
<td>23.58</td>
</tr>
<tr>
<td>Figure 33</td>
<td>11.32</td>
<td>22.64</td>
<td>42.09</td>
<td>22.34</td>
<td>33.62</td>
<td>23.01</td>
</tr>
<tr>
<td>Figure 34</td>
<td>12.00</td>
<td>23.99</td>
<td>29.72</td>
<td>23.21</td>
<td>22.86</td>
<td>24.06</td>
</tr>
<tr>
<td>Figure 35</td>
<td>6.67</td>
<td>17.40</td>
<td>41.86</td>
<td>17.69</td>
<td>23.21</td>
<td>18.03</td>
</tr>
<tr>
<td>Figure 36</td>
<td>9.35</td>
<td>18.69</td>
<td>37.84</td>
<td>23.91</td>
<td>29.58</td>
<td>23.26</td>
</tr>
<tr>
<td>Figure 37</td>
<td>7.37</td>
<td>14.56</td>
<td>29.10</td>
<td>14.73</td>
<td>20.90</td>
<td>15.59</td>
</tr>
<tr>
<td>Average</td>
<td>10.50</td>
<td>21.84</td>
<td>35.57</td>
<td>22.07</td>
<td>26.78</td>
<td>22.77</td>
</tr>
</tbody>
</table>

Scaled histogram equalization and scaled fuzzy logic methods achieved the highest values of sharpness on average. The lower values with homomorphic filtering and single scale retinex are understandable, given that they tend to reduce shadowing in an image. Therefore, while these techniques make hidden features visible, they do so by reducing the general contrast of shadow regions compared to the rest of the image. Therefore, while reducing the sharpness and contrast globally, they are actually enhancing the contrast locally.

7.4 Histogram Standard Deviation

The image histogram standard deviation is used as one of the deterministic metrics in the analysis of this investigation. The higher the value of standard deviation, the greater the histogram is spread across the dynamic range for that image. Thus a higher
value of standard deviation implies a better contrast for the image. Standard deviation of original test images and the scaled output images is calculated and presented here.

The following table depicts the calculated standard deviation of the global histograms for the full test images. This metric provides a view of the overall contrast in the image.

*Table 6 Standard Deviation of Test Images*

<table>
<thead>
<tr>
<th>Image</th>
<th>Original Image</th>
<th>Scaled Original Image</th>
<th>Scaled Histogram Equalization</th>
<th>Scaled Homomorphic Filtering</th>
<th>Scaled Fuzzy Logic</th>
<th>Scaled Single Scale Retinex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 30</td>
<td>8.46</td>
<td>8.68</td>
<td>9.97</td>
<td>9.97</td>
<td>8.00</td>
<td>8.66</td>
</tr>
<tr>
<td>Figure 31</td>
<td>10.42</td>
<td>11.46</td>
<td>10.82</td>
<td>10.82</td>
<td>11.51</td>
<td>11.48</td>
</tr>
<tr>
<td>Figure 32</td>
<td>8.90</td>
<td>10.04</td>
<td>10.72</td>
<td>10.72</td>
<td>10.41</td>
<td>10.33</td>
</tr>
<tr>
<td>Figure 33</td>
<td>5.39</td>
<td>8.79</td>
<td>10.98</td>
<td>10.98</td>
<td>9.05</td>
<td>8.57</td>
</tr>
<tr>
<td>Figure 34</td>
<td>8.72</td>
<td>9.74</td>
<td>10.88</td>
<td>10.88</td>
<td>9.32</td>
<td>9.71</td>
</tr>
<tr>
<td>Figure 35</td>
<td>8.33</td>
<td>8.43</td>
<td>10.53</td>
<td>10.53</td>
<td>8.22</td>
<td>8.42</td>
</tr>
<tr>
<td>Figure 36</td>
<td>7.80</td>
<td>9.50</td>
<td>10.89</td>
<td>10.89</td>
<td>9.89</td>
<td>9.55</td>
</tr>
<tr>
<td>Figure 37</td>
<td>11.80</td>
<td>11.83</td>
<td>10.83</td>
<td>10.83</td>
<td>11.84</td>
<td>11.83</td>
</tr>
<tr>
<td>Figure 38</td>
<td>5.87</td>
<td>9.94</td>
<td>10.59</td>
<td>10.59</td>
<td>10.28</td>
<td>10.02</td>
</tr>
<tr>
<td>Figure 39</td>
<td>9.16</td>
<td>11.39</td>
<td>10.82</td>
<td>10.82</td>
<td>11.56</td>
<td>11.64</td>
</tr>
<tr>
<td>Figure 40</td>
<td>9.09</td>
<td>11.69</td>
<td>10.64</td>
<td>10.64</td>
<td>11.77</td>
<td>11.82</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>8.54</strong></td>
<td><strong>10.13</strong></td>
<td><strong>10.70</strong></td>
<td><strong>10.70</strong></td>
<td><strong>10.17</strong></td>
<td><strong>10.18</strong></td>
</tr>
</tbody>
</table>

Additionally, the local histogram standard deviation was calculated for subset image regions that had challenging test condition. Three specific cases were evaluated: details hidden in a shadow, a region with mid-range brightness and many objects, and a region near to saturation. The following test images were used for this purpose.

![Image (a)](image1.png) ![Image (b)](image2.png) ![Image (c)](image3.png)
Figure 29 Metric Masks: (a) Original test image – high contrast (b) Image with mask extracted from (a) (c) Mask used for metric calculation (d) Original test image – normal contrast (e) Image with mask extracted from (d) (f) Mask used for metric calculation (g) Original test image – details in shadow (h) Image with mask extracted from (g) (i) Mask used for metric calculation

Figure 29 shows the mask only for the original test images. Similar masks were chosen for individual images for scaled original images, scaled histogram equalization images, scaled homomorphic filtering images, scaled fuzzy logic images and scaled single scale retinex images.

The following table depicts the calculated local histogram standard deviation for the mask regions in the challenging test conditions listed above:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>High Contrast Figure 29(a)</td>
<td>12.20</td>
<td>13.26</td>
<td>12.34</td>
<td>13.50</td>
<td>14.24</td>
<td>13.21</td>
</tr>
<tr>
<td>Normal Contr. Figure 29(d)</td>
<td>9.25</td>
<td>10.11</td>
<td>9.15</td>
<td>10.30</td>
<td>8.62</td>
<td>10.18</td>
</tr>
<tr>
<td>Details in Shadow Figure 29(g)</td>
<td>5.16</td>
<td>7.50</td>
<td>9.14</td>
<td>5.39</td>
<td>3.40</td>
<td>7.05</td>
</tr>
</tbody>
</table>
Scaled fuzzy logic achieved the highest histogram standard deviation for an already high contrast image region. However, it performed the worst for providing contrast to details lost in a shadow. Scaled homomorphic filtering and single scale retinex generate the highest contrast or midrange detail in a normal contrast image region. Scaled histogram equalization surprisingly generated better results compared to other methods for details in shadows in the particular image chosen.

7.5 Image Results

The techniques implemented in this investigation were applied to the following images. The test results along with the respective histograms are as shown in the following figures. The test image data set was carefully chosen to include images that have different ranges of contrast as well as challenging conditions. Images with normal, low and high contrast were chosen along with the images that have details in shadow.
Figure 30 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e)

Figure 30 is a high contrast image. The lamp and the fan are the only details in the image with the background being a plain ceiling. The output of scaled histogram equalization as shown Figure 30(e) has a white washed effect near the fan lamps. The brilliance from the light bulbs is prominent while poor contrast is noted along the fan. In the scaled homomorphic filtering and scaled fuzzy logic outputs, the fan can be seen prominently whereas the scaled fuzzy output hides the fan part in the output image. Thus a significant loss of information can be seen in the scaled fuzzy logic output.
Figure 31 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e)

Figure 31 is a medium contrast test image with a lot of details and objects that are enhanced after the implementation of the contrast enhancement techniques. For this test image scaled homomorphic filtering and scaled single scale retinex provide a crisp output image without loss of details.
Figure 32 Image Results: (a) Original Image Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e)

Figure 32 serves as a low contrast image with few details in the test image. In the scaled histogram equalization and scale fuzzy logic outputs, right hand corners the output image are darker and thus do not provide more contrast in that region. Whereas for the scaled homomorphic filter and scaled single scale retinex outputs, better contrast is obtained in the entire output image. These techniques also succeeded at bringing objects and details out of the shadows.

Below are some examples of other high contrast, low contrast and medium contrast images.
Figure 33 Image Results: (a) Original Image  (b) Histogram of (a)  (c) Scaled original image  (d) Histogram of (c)  (e) Scaled histogram equalization applied  (f) Histogram of (c)  (g) Scaled homomorphic filtering applied  (h) Histogram of (e)  (i) Scaled fuzzy logic applied  (j) Histogram of (g)  (k) Scaled single scale retinex applied  (l) Histogram of (e)
Figure 34 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e)
Figure 35 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e)
Figure 36 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e)
Figure 37 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e)
Figure 38 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e)
Figure 39 Image Results: (a) Original Image (b) Histogram of (a) (c) Scaled original image (d) Histogram of (c) (e) Scaled histogram equalization applied (f) Histogram of (c) (g) Scaled homomorphic filtering applied (h) Histogram of (e) (i) Scaled fuzzy logic applied (j) Histogram of (g) (k) Scaled single scale retinex applied (l) Histogram of (e)
These image examples again show similar performance between homomorphic filtering and the single scale retinex methods; with both these methods achieving better results than global histogram equalization. In general, the contrast and average brightness results with homomorphic filtering make it subjectively easier to pick out objects in the image than the SSR images.

The very low contrast images, such as Figures 36-38, demonstrate that while histogram equalization increases the contrast in these images, it also greatly increases the noise in the image. Therefore, while quantitatively better in contrast levels and greatly “improved” in sharpness, these images are subjectively inferior and less useful than those enhanced by other techniques that scored lower on sharpness.
8. CONCLUSION AND FUTURE WORK

Based on the metrics comparisons, scaled histogram equalization and scaled fuzzy logic methods have high values of sharpness on average. This feature might be important in security related applications where precise data is required. Scaled histogram equalization and scaled homomorphic filtering achieve the highest values of global contrast on average, as measured by histogram standard deviation. The contrast achieved is more usable with the homomorphic filtering, however; as the histogram equalization tends to keep objects obscured in shadows and enhances noise in low contrast images. Scaled fuzzy logic and scaled single scale retinex have poorer standard deviations compared to the other implemented techniques. Based on user observations, scaled homomorphic filtering was selected by most users in terms of overall better contrast without outputting a grainy or a white-washed effect. This subjective assessment, along with the apparent ability of homomorphic filtering to bring objects and details out of shadows, makes it well suited to most IR imaging applications such as night vision, firefighting, and security surveillance.

To the existing methods, post-processing can be investigated in the future. This could apply global contrast enhancement methods or local contrast enhancement methods based on the application. The same techniques might also be tested with color infrared images.
REFERENCES


83


APPENDIX A

Software Configuration Procedure

The following procedure was performed after installing Coyote, Photon GUI, and Filter Driver software onto the PC. After this procedure is complete, images may be captured and saved using Image Capture Procedure.

1. **Launch Coyote Application** using the shortcut on the desktop, or in the start menu under Indigo Systems inc. The Application Options dialog will show by default.

2. Click **OK**.
3. The Network Device Finder dialog will display. Select **Filter Driver** and click **OK**.

![Network Device Finder](image)

a. If this is the first time running Coyote since the iPORT Ethernet Module was powered on, the Set IP Address dialog will display with a default IP Address assignment. In this case, Click **OK** in the Set IP Address dialog, then click **OK** in the Network Device Finder dialog.

![Set IP Address](image)

4. The Select Camera dialog will appear. Select **Indigo Systems Camera**, and click **OK**.

![Select Camera](image)
5. In the Coyote window, Device tab, IP Information pane, select **UDP over Filter Driver** for the Communication mode.

6. **Close Coyote.**
7. Open **Photon GUI** from the shortcut on the desktop, or in the start menu under *Indigo*.

8. **Click** the pane in the lower right of the screen, at the far right of the status bar (labeled “iPORT2” in the figure of Step 7). The Select Communication Parameters dialog will open.
9. From the drop-down list, select the **Filter** item, and click **OK**.

10. In the **Digital Video** tab, do the following:
   b. In Host-based AGC Parameters pane, select **Auto-adjust display dynamic range**.
   c. Click **Start Video**.
   d. Click **Stop Video**.

11. Close **Photon GUI**.

12. **Launch Coyote Application** by repeating Step 1 through Step 5.

13. From the **Acquisition** tab, click **Configure**.
14. The Configuration – Advanced dialog will appear. From the Image Settings tab, Windowing pane, set **Width** to 320 and **Height** to 240.

![Configuration - Advanced dialog](image)

15. From the **Pixel Type** tab, do the following:

   a. For Color space, select **Grayscale**.

   b. For Pixel Depth, select **8 bits**.
16. Click **OK**.

The camera is now configured and ready to capture and save images.

**Image Capture Procedure**

After performing Software Configuration Procedure, the following procedure was followed to capture and save images.

1. Navigate to **Acquisition** tab within **Coyote**. Refer to figure in Software Configuration Procedure, Step 13.
2. In Frame save pane, do the following:
   
a. Select **Windows Bitmap** radio button.

   b. Click **Save To…** button and enter desired image save location.

3. In Acquisition Control pane, select **Grab - Single** from the dropdown list.

4. In Frames to save pane, enter a number of frames to save (per run), as desired.

5. Click **Save** to begin a capture run.

6. Direct camera as desired, then click **Start** to capture image. Repeat for as many as were entered in Step 4. (A progress bar will appear in Frame save panel to show how many images have been captured out of the number entered in Step 4.)

7. Repeat Step 4 through Step 6 as desired.
% This code is for implementing homomorphic filtering contrast enhancement technique
clc
close all
clear all
tic
d=10;

im=double(imread('14.bmp'));
[r c] = size(im);

A= zeros(r,c);
for i=1:r
    for j = 1:c
        A(i,j) = (((i-r/2).^2+(j-c/2).^2)).^(0.5);
        B(i,j) = A(i,j)*A(i,j);
        H(i,j) = (1-exp(-((B(i,j)).^2/d.^2)));
    end
end

rL= 0.0999;
rH=1.01;
H = ((rH-rL).*H)+rL;
im_l = log(im+1);
im_f = fft2(im_l);
im_nf = H.*im_f;
im_n = abs(ifft2(im_nf));
im_e = exp(im_n)-1;
imshow((im_e),[])
toc
APPENDIX C

%This code is for implementing fuzzy logic contrast enhancement technique
function [output] = FuzzyTransform(input)

%FUZZIFY
dark = Sigmoid(127, 0, input);          %(low, high, input)
midGrey = Gaussian(127, 127, input);    %(width, shift, input)
light = Sigmoid(128, 255, input);       %(low, high, input)

%INFER (fuzzy rules, trivial since one input variable)
%If dark then darker
%darker = dark
%If mid-grey then mid-grey
%mid-grey = midGrey
%If light then lighter
%lighter = light

%DEFUZZIFY (CENTROID)
darker_center = 0;
midGrey_center = 127;
brighter_center = 255;
output = (dark * darker_center + midGrey * midGrey_center ...
    + light * brighter_center) / (dark + midGrey + light);
end

%Implements a Sigmoid / Logistic function with specified 1% and 99% point.
%  T1 is 10% point
%  T2 is 90% point

function y = Sigmoid(T1, T2, x)
    shift = 0.5 * (T1 + T2);
    a = 9.2 / (T2 - T1);
\[ y = \frac{1}{1 + \exp(-a(x - \text{shift}))}; \]

\%W is the width of the gaussian bell curve, measured from its 0.01 points.
\%shift is the center point.

\textbf{function} y = Gaussian(W, \text{shift}, x)
\begin{align*}
a &= \sqrt{W^2/4.60}; \\
y &= \exp(-((x - \text{shift})/a)^2);
\end{align*}
\textbf{end}

\%Plot classes / membership functions
\begin{verbatim}
x = 0:255;
for i = 1:256
    dark(i) = Sigmoid(127, 0, x(i));  \%low, high, input
    midGrey(i) = Gaussian(127, 127, x(i));  \%width, shift, input
    light(i) = Sigmoid(128, 255, x(i));  \%low, high, input
end
\end{verbatim}

figure
hold on
plot(x, dark)
plot(x, midGrey)
plot(x, light)

clear all
close all
imgIn=double(imread('5.bmp'));
subplot(121)
imshow(imgIn./255);
for r = 1:240
    for c = 1 : 320
        imgOut(r,c) = FuzzyTransform(imgIn(r,c));
    end
end
subplot(122)
imshow(imgOut./255)
%This code is for implementing single scale retinex contrast enhancement technique

function [L,R] = new_ssr1(S, alpha, beta)
S=cast(S, 'double');
maxS= max(max(S));
minS = min(min(S));
S = (255/(maxS-minS))*(S-minS);
s= log(S+1);
l=s;
dt=0.01;
n=100;
for i = 1:n
    G= -(4.0*del2(l))+alpha*(l-s)-beta*(4.0*del2(l-s));
    l=max(l-dt*G, s);
end
L=exp(l);
L=cast(L, 'double');
R=exp(s-1);
R=cast(R, 'double');
subplot(122)
imshow ((R), []);
close all
end

tic
im=double(imread('27.bmp'));
subplot(121)
imshow(im./255);
new_ssr1(im,90,100);
toc
APPENDIX E

%This code is for computing the sharpness of the images

function out = sharpness(imPath)

%Configuration
if nargin == 0  % if no input arguments, set default path
    imPath = '14_FzL.bmp';
end

prew_Vert = fspecial('prewitt');
I = double(imread(imPath))./256;  % imfilter requires double
grad_Vert = imfilter(I, prew_Vert);
grad_Horiz = imfilter(I, prew_Vert');
grad_Mag = sqrt(grad_Vert.^2 + grad_Horiz.^2);

%sharpness
out = mean(mean(grad_Mag))*256;
end
%%This code is for computing the histogram standard deviation of the images

%%Input image paths
inPath0 = '9.bmp';
inPath1 = '9_scaled.bmp';
inPath2 = '9_HE_scaled.bmp';
inPath3 = '9_HmF_scaled.bmp';
inPath4 = '9_FzL_scaled.bmp';
inPath5 = '9_SSR_new_scaled.bmp';

%%Read images
I0 = imread(inPath0);
I1 = imread(inPath1);
I2 = imread(inPath2);
I3 = imread(inPath3);
I4 = imread(inPath4);
I5 = imread(inPath5);

%%Identify mask
mask0 = roipoly(I0);
mask1 = roipoly(I1);
mask2 = roipoly(I2);
mask3 = roipoly(I3);
mask4 = roipoly(I4);
mask5 = roipoly(I5);

%%Isolate only masked pixels (to a column vector)
px0 = I0(mask0);
px1 = I1(mask1);
px2 = I2(mask2);
px3 = I3(mask3);
px4 = I4(mask4);
px5 = I5(mask5);

%%Output standard deviations
stdev0 = std(px0);
stdev1 = std(px1);
stdev2 = std(px2);
stdev3 = std(px3);
stdev4 = std(px4);
stdev5 = std(px5)