2006-12

Fusion of night vision and thermal images

Neo, Tiong Tien

Monterey California. Naval Postgraduate School

http://hdl.handle.net/10945/2489

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

FUSION OF NIGHT VISION AND THERMAL IMAGES

by

Tiong Tien Neo

December 2006

Co-advisors: Monique P. Fargues
Alfred W. Cooper

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2006	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Fusion of Night Vision and Thermal Images	**5. FUNDING NUMBERS**
6. AUTHOR(S) Tiong Tien Neo	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Night vision and thermal images are extensively used in military operations, as they help in mission planning and executions tasks. Image fusion effectively combines information present in each type of image. This research explored two wavelet-based image fusion approaches for night vision and thermal images; namely wavelet transform fusion and region-based fusion. Morphological methods designed to improve the image segmentation step were considered to improve image contrast and a global image quality index was applied to investigate the information content improvement resulting form the fusion process. Finally, a MATLAB-based graphical user interface was designed to assist the user in evaluating the benefits of the fusion process. Results showed the selection process is able to narrow to the best fused image with a satisfactory accuracy.

.

14. SUBJECT TERMS Image Intensifiers, Thermal Image, Digital image processing, Wavelet transform fusion, Region-based fusion	**15. NUMBER OF PAGES** 133
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

FUSION OF NIGHT VISION AND THERMAL IMAGES

Tiong Tien Neo
Captain, Republic of Singapore
B.Eng (Electrical), University of New South Wales (Australia), 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMBAT SYSTEMS SCIENCE AND TECHNOLOGY

from the

NAVAL POSTGRADUATE SCHOOL
December 2006

Author: Tiong Tien Neo

Approved by: Monique P. Fargues
Thesis Co-Advisor

Alfred W. Cooper
Thesis Co-Advisor

James H. Luscombe
Chairman, Department of Physics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Night vision and thermal images are extensively used in military operations, as they help in mission planning and execution tasks. Image fusion effectively combines information present in each type of image. This research explored two wavelet-based image fusion approaches for night vision and thermal images namely wavelet transform fusion and region-based fusion. Morphological methods designed to improve the image segmentation step were considered to improve image contrast and a global image quality index was applied to investigate the information content improvement resulting form the fusion process. Finally, a MATLAB-based graphical user interface was designed to assist the user in evaluating the benefits of the fusion process. Results showed the selection process is able to narrow to the best fused image with a satisfactory accuracy.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1
  A. MILITARY NEEDS 1
  B. IMAGERY FUSION 1
  C. LIMITATIONS 2
    1. Device Level Fusion 3
    2 Computational Image Fusion 4
  D. OBJECTIVES 4

II. BACKGROUND 7
  A. IMAGERY SENSOR SYSTEMS 7
    1. Image Intensifier 8
      a. Gen 0 8
      b. Gen 1 9
      c. Gen 2 9
      d. Gen 3 10
      e. Gen 4 11
    2. Thermal Imager 12
      a. Theoretical Background 12
      b. Detectors and Electro-Optics System 15
      c. Performance Criteria 18
  B. IMAGE FUSION 20

III. IMAGE FUSION METHODS 21
  A. WAVELET TRANSFORMATION FUSION 21
    1. Wavelet Transform 22
    2. Discrete Wavelet Transform: Implementation to Image Fusion 22
    3. Wavelet Families 25
    4. Image Fusion 29
      a. Image Resampling 29
      b. Image Registration 29
    5. Fusion Schemes 30
      a. Fusion Rule 1 – Selection of Dominant Mode 33
      b. Fusion Rule 2 – Weighted Average of Modes 33
      c. Fusion Rule 3 – Weighted Average of Window-based Modes 34
  B. REGION- BASED FUSION 34
    1. Region Segmentation 35
    2. Watershed Transform 35
    3. Fusion Methodology 37
      a. Segmentation Process and Feature Activity Information 38
      b. Fusion Rule 39

IV. DIGITAL IMAGE PROCESSING 41

A. IMAGE CHARACTERISTICS 41
1. Composition 41
a. Night Vision 42
b. Thermal Image 44
2. Atmospheric Effects on Image Acquisition 46
B. IMAGE ENHANCEMENT 48
1. Image Histogram 49
2. Contrast Limited Adaptive Histogram Equalization 50
3. Experimentation of CLAHE with Night Vision and Thermal Images. 51
4. Observations and Comments 51
D. FEATURES AND OBJECTS 55
1. Intensity Identification 56
2. Edge Enhancement 56
3. Color Depiction 56
E. IMAGE QUALITY EVALUATION 57

V. IMAGE FUSION APPLICATION 61
A. APPLICATION PROCESS FLOW 61
B. GUI APPLICATION 63
1. Image Selection and Execution of Fusion 63
2. Information Display 64
3. Wavelet Decomposition Process 64
4. Image Overlays 65
C. UNDERLYING PROCESSES 66
D. RESULTS AND DISCUSSION 67

VI. CONCLUSION AND RECOMMENDATIONS 69
A. SUMMARY 69
B. RECOMMENDATION FOR FURTHER RESEARCH 69
1. Counterchecking of Fusion Application with Human Visual Evaluation 69
2. Other Image Quality Evaluation Methods 70
3. Real- Time Implementation 70

APPENDIX A. MATLAB ALGORITHMS 71

APPENDIX B: DETAIL GUI APPLICATION FLOWCHART 105

APPENDIX C. GUI EXPERIMENT RESULTS 107

LIST OF REFERENCES 113

INITIAL DISTRIBUTION LIST 117

LIST OF FIGURES

Figure 1. Electromagnetic Spectrum Showing the Wavelengths of Ultra-violet, Visible Light and Infrared. (From: Science @ Nasa, Oct 2006, Electromagnetic Spectrum. Retrieved October 22, 2006 from http://science.nasa.gov/newhome/help/glossfig1.hm) 2

Figure 2. Timeline of image intensifiers 8

Figure 3. Image Intensifier Internal Construction and Mechanism. (From: American Technologies Network Corporation, Oct 2006)................... 10

Figure 4. Wien Displacement Law, at a Specific Temperature the Peak Emission Wavelength is the Corresponding Value. (From: Brooker, 2006) 14

Figure 5. Read Out Mode of a Focal Plane Array (From: Graham, 2006) 17

Figure 6. Second-generation PV HgCdTe Detectors, Which are Configured in (a) Parallel Scan with Time Delay Integration (TDI) and (b) Staring (parallel) Sensor Formats (From: Norton, 2002)................................ 17

Figure 7. Fusion of the Wavelet Transforms of Two Images (From: Nikolov et al., 2002) 21

Figure 8. Subbands Representation in a Multi-resolution Decomposition of an Image 24

Figure 9. Decomposition Tree of the Original Images into Four Subband Images. (After: Chow 2004)... 24

Figure 10. Reconstruction Tree from 4 Subband Images Back to Recover the Original Image. (After: Chow 2004) 24

Figure 11. Wavelet Families (From: Mathworks, 2006) 26

Figure 12. Experiment for Edge Detection of Wavelet Families........................... 28

Figure 13. General framework for wavelet transforms fusion. (After Chow, 2004) 30

Figure 14. Framework for the Formation of the Fusion Decision Map. (After Chow, 2004) 31

Figure 15. Region Segmentation of the Approximate NVD Image at Three Levels of Decomposition: a) Level 1, b) Level 2 and c) Level 3. (After: Chow, 2004) 36

Figure 16. Region Segmentation of the Approximate Thermal IR Image at Three Levels of Decomposition: a) Level 1, b) Level 2 and c) Level 3. (After: Chow, 2004) 36

Figure 17. Framework for the Formation of the Fusion Decision Map for Region-based Fusion. (After: Chow, 2004) 37

Figure 18. Region Segmentation: a) Region Representation of Image A; b) Region Representation of Image B and c) Joint Region Map, Indicating the Four Identified Regions. (From: Chow 2004) 38

Figure 19. Night Vision and Thermal Images Examples (a) Night vision (From: Military, 2006), (b) Black Hot, (c) White Hot and (d) Thermo-flash Color Vision. (From: imaging1, 2006) 41
Figure 20. Soldiers at Foreground Clearly Identifiable, Light Sources and Helicopter at Background not as Differentiable.(From: Military. 2006) 42
Figure 21. Flare Effect of Bright Lights on the Ships (Photograph courtesy of Naval Research Laboratory).. 43
Figure 22. (a) Exhibiting an Overall Darkness and (b) Exhibiting an Overall Brightness. (Photographs courtesy of Naval Research Laboratory)... 44
Figure 23. Examples of Color Display of Thermal Images. (From: Infrared1, 2006) 44
Figure 24. The atmospheric transmittance of electromagnetic radiation (From: Jane's August 2005).. 47
Figure 25. Atmospheric Effects : Absorption and Scattering on EM radiation (From: Guyot, 1997) 48
Figure 26. Histogram of Four Basic Image Types. (From: Teo, 2003)................. 50
Figure 27. Comparison of Original and CLAHE Processed for Scene A 52
Figure 28. Comparison of Series of Fused Images for Scene A.......................... 53
Figure 29. Comparison of Original and CLAHE Processed for Scene B 54
Figure 30. Comparison of Series of Fused Images for Scene B.......................... 55
Figure 31. Overlays Construction Process 57
Figure 32. General Fusion Process Application Flow 62
Figure 33. Screenshot of GUI Application with Fused Image and Overlays 65
Figure 34. a) Threshold Adjustment, b) Loop Process Flowchart........................ 66
Figure 35. Experiment Test Results for GUI Application. 68

LIST OF TABLES

Table 1. Comparison of Generation of Image Intensifiers. (After: Montoro H.P, Oct 2006; ATN Corp, Oct 2006) 11
Table 2. Comparison of Night Vision and Thermal Images. 45

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to extend my most sincere thanks and gratitude to Professor Monique P. Fargues, Electrical and Computer Engineering Department, Naval Postgraduate School for her patience, guidance and her invaluable advice during the completion of this thesis. I thank her for the time she has spent imparting her knowledge and for allowing me to explore and apply the concepts I acquired on wavelet transform fusion. I would also like to thank Professor Alfred W. M Cooper, Physics Department, Naval Postgraduate School for his valuable advice and support on the theoretical background required in this thesis.

I am the most grateful to my wife Vikki, for her endless love, support, encouragement and understanding that has helped me immensely in the completion of my Master’s program and during this thesis process. I am also grateful to my parents for their constant support and encouragements during my studies.

Finally, I would like to acknowledge the support of the Ministry of Defense, Singapore, for giving me the opportunity to pursue this Master’s degree in the United States.

This research was performed in conjunction with Project Reference Number TDSI /02-010/A supported by the Temasek Defense Technology Institute, National University of Singapore.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MILITARY NEEDS

Modern warfare reliance on information and imagery has been exemplified by the recent war in Iraq (Operation Iraqi Freedom) and the Israel-Hezbollah conflict. Commanders on the ground need up-to-date imagery information to get an accurate appreciation of the battlefield as well as to assist in the planning and conducting of ground operations. Often this information can be obtained through various agencies and units. Imagery Intelligence (IMINT) is the gathering of military intelligence via the use of satellite or aerial imagery (Wikipedia, 9 Oct 2006). In recent conventional warfare, this has been translated to the use of unmanned aerial vehicles (UAV) or other forms of manned and unmanned means for the collection of timely visual reports on the enemy. Information from imagery is useful and simplifies the complexity of having ground troops updating via sighting reports or verbal reports via voice communications, which frequently is inadequate to provide information that would have impacts on the outcome of operations.

B. IMAGERY FUSION

In recent times, multi-spectral imagery systems have been used extensively for area surveillance and intelligence collection. Multi-spectral imagery systems help to provide additional information and allow for the capture of imagery under imperfect conditions, such as low light and foggy conditions that may not favor conventional imagery systems which capture imagery in the visible light spectrum. They also range from sophisticated systems installed in a satellite for large area imagery intelligence to hand-portable devices used by soldiers in the battlefield for situational awareness, and its applications are highly versatile.

Conventional visible light images are taken with three colors, whereas multi-spectral images explore other frequencies in the electromagnetic (EM) spectrum from visible light as well as frequencies close to visible light such as infrared(IR). A Multi-spectral image allows the extraction of additional information that a conventional visible light image fails to capture. Recent developments include researches to explore the ultra-violet regions of the electromagnetic spectrum. Figure 1 shows the various regions of the electromagnetic spectrum.

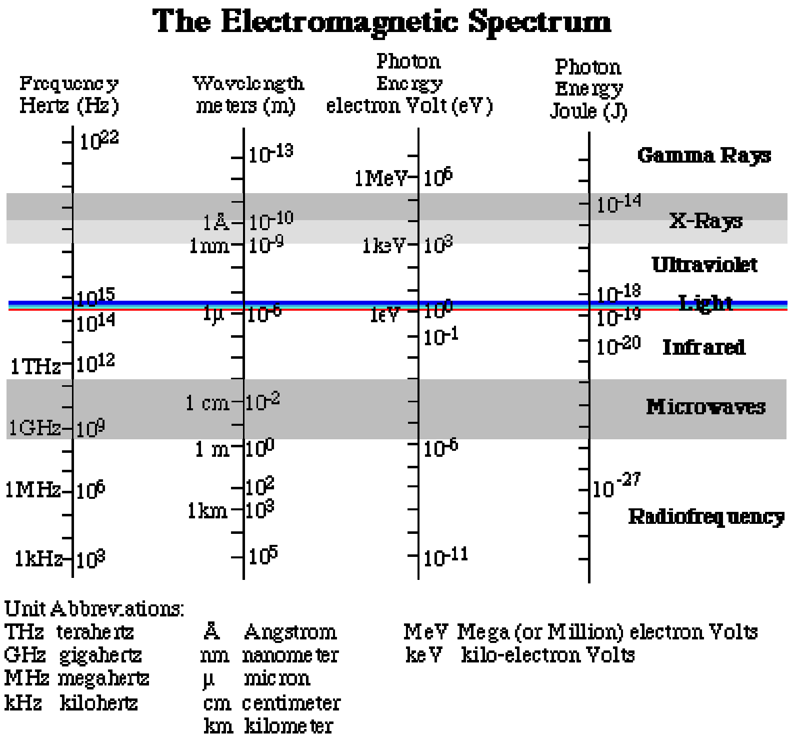


Figure 1. Electromagnetic Spectrum Showing the Wavelengths of Ultra-violet, Visible Light and Infrared. (From: Science @ Nasa, Oct 2006, Electromagnetic Spectrum. Retrieved October 22, 2006 from http://science.nasa.gov/newhome/help/glossfig1.hm)

However, most imaging systems still only have the capability to detect wavelengths from a specific region of the EM spectrum. Often the information provided by a particular sensor is complementary to another if both images are viewed as one and any drawbacks in one sensor can be filled in by another. So by combining imagery from multiple sensors, an improved imagery can be conceived to help in providing completeness to the information required.

C. LIMITATIONS

To implement fusion of imagery on devices, the fusion method can be applied at a selected level of implementation. Research on implementation at the device level is currently being carried out by industry as well as academia and involves fusion of detectors to enable multi-spectral detection, which at the current stage might not be realizable for night vision and IR devices. However, most adaptation of device level fusion is implemented through digital fusion at the device software algorithm level as this approach offers fewer technical difficulties than designing detectors with multi-spectral detection capability. Another form of fusion focuses on digitally processing the image on a separate machine, by manipulating the images captured and using a fusion scheme to digitally combine the images.

1. Device Level Fusion

Current limitations of multi-spectral imagery lie primarily on the ability to combine various images that are captured from different regions of the spectrum of visible light, Near-IR and Mid-wave IR at the detectors. Image fusion cannot be done at the origin of the imagery by simply overlapping the detectors from two types of systems, as detectors may be designed differently based on the behavior of the electromagnetic radiation at a particular wavelength. In addition, different materials are used to detect different wavelengths, and to fabricate a detector capable of detecting a range of wavelengths using several materials on a single piece of material itself is a manufacturing challenge considering that a typical detector array is about 2 to 3 square centimeters. Furthermore, the finite size of the detectors makes it impossible to incorporate two types of material on one piece of detector and still be able to function well, unless a specific type of material that can be used to detect both wavelength from the visible light and the IR regions exists.

The imageries of interest in this thesis are Night Vision Device (NVD) and Thermal Imager (TI) images. NVD capture images through the amplification of tiny amounts of light through a system of photon-electron conversion and electron amplification. TI uses detectors with electromagnetic sensitive semi-conductor materials to detect IR radiation. The details on the technology involved in the two devices will be discussed in the next chapter. Since the image acquisition methods from the NVD and TI are different, it is a challenge to integrate them into one single system. Another possibility in developing a fully integrated NV/TI system may involve changing the method of acquiring night vision images, which itself is a hurdle to overcome, since NVD development has come a long way since WWII. Northrop Grumman Electro-Optical Systems (EOS) has implemented digital fusion in the development of advanced night vision goggles for the US Army. Advanced night vision goggles incorporate conventional image intensified night vision goggles with thermal infrared imaging to provide a digitally fused image for direct viewing. (Northrop Grumman, Oct 2003). In this implementation the imagery is automatically fused digitally, before the final image is presented to the user in near real time. The advantage of this implementation is that it allows the information to be presented almost immediately with additional information over conventional NVGs. Device level image fusion does not require the user to carry out any digital image processing. However, depending on how the NVG and TI systems are integrated, the device may be too cumbersome to be designed as a hand-held device.

2 Computational Image Fusion

Image fusion may involve manipulating the original captured images from different devices and combining them. It may also involve adjusting image attributes for further processing prior to fusion. Often these processes require high speed computers, which are currently too bulky to be incorporated as hand-portable devices. Where time is paramount in operations, the possibility of getting the image processed separately may be deemed inefficient or even useless.

D. OBJECTIVES

This thesis is an extension from two other previous theses completed in December 2003 and December 2004. First, Chek Koon Teo’s studies on the digital enhancement of night vision and thermal images (Teo, 2003) focused on improving the region segmentation step needed prior to image fusion. Next, Khin Choong Chow explored possible fusion methods and experimented with different schemes in his thesis titled “Fusion of images from dissimilar sensor systems”

(Chow, 2004). The result from this thesis proved helpful in conceptualizing the methodology for fusion of night vision and thermal images and also in designing the GUI application. This thesis attempts to capitalize on the two theses by

- Proposing configurations to determine the optimal settings for both pixel-level and region-based fusion.
- Identifying suitable applications so that the fusion rules can be automated.
- Proposing an image evaluation to be incorporated in the application.

In Chapter II, the theoretical background for night vision and thermal imagery is discussed, highlighting some limitations pertinent to acquiring these images. In Chapter III, the two fusion schemes used in this thesis are examined. Chapter IV describes the method and the experiments with the digital image processing required before the images are fused. Chapter V describes the fusion process and the GUI application. Finally, conclusions and recommendations are presented in Chapter VI.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

This chapter will discuss the development of night vision devices and thermal imagers since they were first introduced. The theoretical background of night vision and thermal image devices is covered to allow an in-depth understanding and appreciation of the challenges and inherent limitations on these devices that are the motivation behind image fusion.

A. IMAGERY SENSOR SYSTEMS

Imagery sensor systems enable commanders to have a better situational awareness of the battlefield and allow information acquisition to be done depending on the needs of the mission planning. In modern warfare, imagery sensor systems are fully integrated into battle scenarios to fully exploit their capabilities. They also have the ability to be employed as part of a network centric warfare system for the purpose of increasing combat power and mission success. (Force Transformation, January 2005). This section will introduce the Night Vision Device (NVD) and the Thermal Imager (TI). The main distinction between NVD and TI lies in the different regions of the electromagnetic spectrum that each device is limited to. NVD is limited to wavelengths under 1.2 micrometers (µm). This wavelength range is not emitted by most objects thermally at normal atmospheric temperature as it requires the objects to be at a temperature of 2500 degree Kelvin to do so. As a result, most available light photons[1] are from sky radiation not scene objects with the exception of tungsten filament light bulbs. This means that most objects at near room temperature only emit at 3µm and above.

[1] Photon – In 1926, the chemist Gilbert Lewis suggested the name “photon” for the “quantum of light” and it has been so identified ever since. (as cited in F.L. Pedrotti, L.S. Pedrotti, & L. M. Pedrotti, 2007, p.4). “In modern physics, the photon is the elementary particle responsible for electromagnetic phenomena. It mediates electromagnetic interactions and makes up all forms of light. The photon has zero invariant mass and travels at the constant speed *c*, the speed of light in empty space.” (Wikipedia, 21 Oct 2006)

1. Image Intensifier

One of the most extensively used imagery sensor systems is the night vision device or image intensifier. From army troopers on the ground to helicopter pilots in the air, the use of night vision devices enables military forces to overcome limitations when conducting operations at night. It also provides the side who utilizes this technology a marked advantage over the one who does not in military operations. However, an image intensifier does have limitations due to the natural phenomena or characteristics of light propagation through the atmosphere and the application of technology in military systems. An NVD amplifies light photons in low light conditions through a series of electron photon conversion. The current NVD technology is in its fourth generation. Figure 2 below shows timeline of the four generations of image intensifiers.

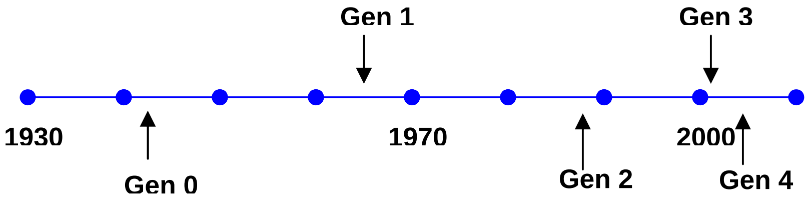


Figure 2. Timeline of image intensifiers

a. Gen 0

The generation zero, or Gen 0, of night vision device was first introduced in World War II where snipers utilized sniper scope during operations. The technology was an evolution from an image converter tube used in television in the mid-1930s developed by an employee of the U.S. Company Radio Corporation of America (RCA). These first generation image devices execute a small amount of light amplification and at the same time enable viewing of images in the IR band. (Brown, Oct 1946) For example, the S1 photocathode used in Gen 0 is made from mixed oxides of silver and caesium (Montoro H.P., Oct 2006) and is not as efficient as today’s systems.

b. Gen 1

In the 1960s, the Vietnam War saw an introduction of the Gen 1 to its operations. Even though its technology was not significantly better than Gen 0, these devices used the photon-electron amplification process, i.e., a series of connected tubes for the amplification of electrons once the photons are converted. Gen 1 devices saw the increase in the amplification gain to 70 with the use of the S20 cathode, which was a significant improvement over Gen 0 devices. However Gen 1 devices had similar drawbacks to Gen 0 products as they did not function well under low light conditions. The S-20 photocathode in Gen 1 devices used a combination of antimony-potassium-sodium-caesium and provided about three times the photo sensitivity of Gen 0 products. However this level of amplification still had limitations under starlight conditions, and problems with overload of the intensifier tube caused by bright lights.

c. Gen 2

The micro-channel plate (MCP) that was introduced in Gen 2 night vision devices increased the amplification significantly and was coupled with further improvements such as the S25 cathode and photocathode processes. Even though Gen 2 devices boosted light amplification between 3,500-10,000 times and resulted in a decrease in the overall device size, the effective gain did not increased significantly. With the introduction of the MCP, the luminance gain was overcome by other losses such as the loss of resolution. (RCA, 1974) The MCP forms the basis of the photocathode amplification in image intensifiers in Gen 2 and later generations' products. Figure 3 shows the internal construction and the mechanism of such a device.

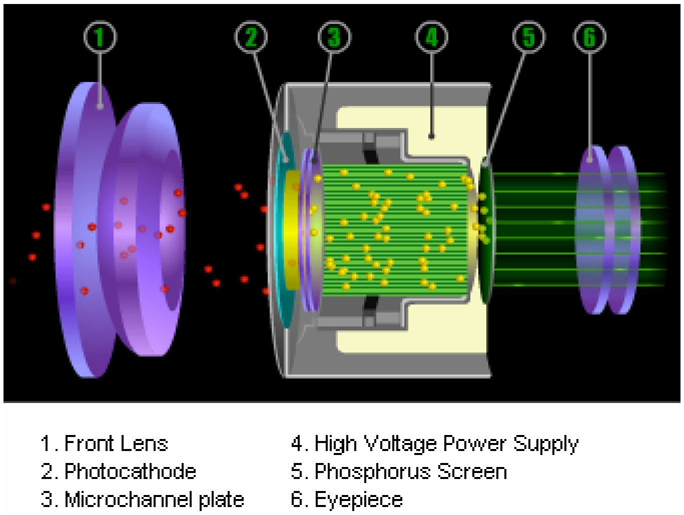


Figure 3. Image Intensifier Internal Construction and Mechanism. (From: American Technologies Network Corporation, Oct 2006)

In Gen 2 devices, light photons enter the image intensifier through the front lens and are collected by the photocathode, where the photons are converted to electrons. Next, the electrons are accelerated through the MCP by the high voltage supply source, resulting in electrons amplified by a factor of 10,000 to 80,000. Amplified electrons are subsequently collected on the phosphor screen where the conversion back to photons takes place. The final photon amplified image is projected through the eyepiece to the user.

d. Gen 3

Gallium arsenide (GaAs) was used as the photocathode material in Gen 3 products which resulted in amplification levels reaching 80,000. GaAs has many advantages over other materials for optical imager type applications. GaAS generates less noise than other types of semiconductor material and is useful in weak signal amplification applications. In addition, GaAs products are highly resistant to radiation damage which makes them ideal for space and military applications. (Logitech, Oct 2006). (Roper Scientific Inc., 2002)

The cost of today's Image intensifiers has come down due to the reduction in production cost of semiconductors in recent years. A typical Gen 3 image intensifier costs about $3,000 per piece whereas Gen 2 products prices averaged around $6,000 when they were first introduced. However, GaAS products are based on a single crystal substrate structure, which is costly and contributes greatly to the overall cost of Gen 3 image intensifiers. In addition, GaAs photocathode products must deal with the degradation of inherent chemical interactions taking place in the MCP due to the electron interactions with the walls of the MCP channels. Thus, a thin metal oxide coating commonly known as an ion barrier film is added to the input side of the MCP to reduce sparkle due to back acceleration of positive ions. (Montoro H.P., Oct 2006)

e. Gen 4

At present, Gen 4 devices exhibit the latest breakthrough in technology with the removal of the ion barrier and the introduction of device gating. This current generation designation has yet to be officially adopted by the U.S military even though many leading manufacturers of image intensifiers have begun to use it for the upgrade from photocathode with film ion barrier to filmless gated tubes. An important improvement for Gen 4 products is a greater target identification range and higher target resolution over Gen 3 devices, particularly at extremely low light conditions (ATN Corp, Oct 2006). This characteristic is particularly important for military operations which require the capability to differentiate friend from foe.

Generation	Gen 0 (Early 1940s)	Gen 1(Mid 1960s)	Gen 2 (Late 1980s)	Gen 3 (Early 2000s)	Gen 4 (Mid 2000s)
Features	Glass/Metal Seals with S-1 Cathode	Glass/Metal Seals with S-20 Cathode	Ceramic/Metal Seals with S-25 Cathode Mircro-channel plate	Ceramic/Metal Seals with GaAS Cathode Mircro-channel plate with ion barrier film	Ceramic/Metal Seals with GaAS Cathode Gated Filmless tubes
Specifications	Gain: >1 SNR: 2	Gain: 70 SNR: 6	Gain: 3500 – 10,000 SNR: 10 - 12	Gain: 80,000 SNR: 19 -21	Gain: 80,000 SNR: 25 - 26
Amplification Conditions	Ambient lights Lux: 0.1	Full moonlight Lux: 0.1 – 0.01	1/4 moonlight Lux: 0.001	Starlight Lux: 0.00001	Limited starlight Lux: 0.00001

Table 1. Comparison of Generation of Image Intensifiers. (After: Montoro H.P, Oct 2006; ATN Corp, Oct 2006)

2. Thermal Imager

The thermal imager is complementary to the image intensifier. Unlike the image intensifier, the history on the development of the thermal imager is not as exciting, partially due to the fact that the thermal imaging had sensitive military applications and was expensive when it first started out. The first thermal imager products were manufactured for the military in the 1960s. The development of the thermal imager centered mainly on the techniques of detecting thermal radiation from the environment using different emerging materials and detector designs.

The first detector technique used in the development of the thermal imagers was the scanning of single element detector to produce line images. The next major development was the micro-bolometer in the late 1980s with its patent awarded in 1994. Subsequent developments focused mainly on incorporating the device into equipment systems such as the drivers' viewer for armored vehicles, weapons sights and hand portability. (Bullard, Oct 2006). Some other improvements focused on the development of large scale focal plane arrays.

a. Theoretical Background

Thermography or thermal imaging is a form of thermal imaging that detects electromagnetic radiation in the thermal IR region, typically between the wavelengths of 3 and 15µm. Other familiar terms to define the regions may be Near IR (NIR – 0.7 to 1µm), Short-wave IR (SWIR – 1 to 3µm), Mid-wave IR (MWIR – 3 to 5µm) and Long-wave IR (LWIR – 8 to 12µm). These terms are defined based primarily on the atmospheric transmittance which affects the ability for the radiation to propagate through the air and be detectable. Other IR regions are greatly attenuated or absorbed by the atmosphere such that the study on these regions does not yield any benefits. Further details will be covered in later sections.

As thermal imaging relies on thermal radiation emitting from targets in the background, it is therefore important to understand how this thermal radiation is generated and how it may be picked up by detectors. Next we introduce the fundamentals of IR electromagnetic radiation.

(1) Thermal signature. The primary motivation for the exploitation of thermal imaging by military forces is the emittance of electromagnetic signatures in the 3-5µm and 8-12µm bands either by reflection of solar radiance or thermal-emittance by military targets, i.e., vehicles, soldiers and equipment. The thermal emittance part of the signature is independent of day or night conditions. At the same time, it is difficult to suppress thermal signatures using conventional methods of military camouflaging and in most cases these signatures are generally higher than that of the environment.

(2) Wien's Displacement Law. The peak emitted spectral radiance is as described by Wien's Displacement Law. German physicist Wilhelm Wien stated that the wavelength carrying the maximum energy is inversely proportional to the absolute temperature of the source black body. This property means that the hotter a body is, the shorter the wavelength will be. This relationship is derived from Planck's blackbody radiation and states that

$$\lambda_{\mathrm{m}} T = 2897.8 \pm 0.4 \mu^{\circ} K \quad (\mu m \times K), \tag{2.1}$$

where λ_{m} is the wavelength of maximum spectral radiant emittance and T is the absolute temperature in degree Kelvin. (Hudson, 1969) For example, at temperature T equal to 300K, the wavelength of maximum intensity λ_{max} is 10 μm. So at a temperature of 30˚C, below which most targets are found, objects would be emitting at a wavelength that falls within the 8-12µm band of the transmission window. The relationship is depicted in Figure 4 below.

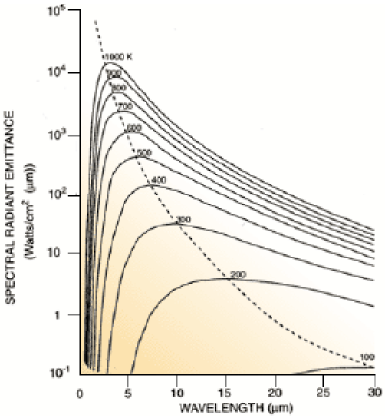


Figure 4. Wien Displacement Law, at a Specific Temperature the Peak Emission Wavelength is the Corresponding Value. (From: Brooker, 2006)

Next, Planck's Law of Blackbody Radiation will be discussed, which is fundamental to understanding how objects emit radiation. This is important as targets that the thermal imager may encounter have a vast range of temperatures and the amount of radiation they emit is dependent on the temperature of the object.

(3) Planck's Law of Blackbody Radiation. Planck's blackbody radiation law describes radiation emission from all bodies that are above absolute zero. Recall that a blackbody is one that absorbs all incident electromagnetic radiation and the Planck's law of blackbody radiation predicts the spectral intensity (or spectral radiance) I of electromagnetic radiation at all wavelengths from a blackbody at temperature T, where I is defined as

$$I(\nu,T)=\frac{2h\nu^3}{c^2}\frac{1}{e^{\frac{h\nu}{kT}}-1}, \quad (J\times s^{-1}\times m^{-2}\times sr^{-1}\times Hz^{-1}) \qquad (2.2)$$

ν is the frequency, h is Planck's constant equal to $6.6260693\times10^{-34}J\cdot s$, k is the Boltzmann constant equal to $1.3806505\times10^{-23}J\cdot K^{-1}$ and c is the speed of light equal to $2.9979\times10^{8}ms^{-1}$. Planck's law also calculates the spectral exitance M of an object, defined as

$$M(\lambda,T)=\frac{2\pi hc^2\lambda^{-5}}{e^{\frac{hc}{\lambda kT}}-1}, \quad (W\times m^{-2}\times \mu m^{-1}) \qquad (2.3)$$

where the parameter h is Planck's constant($6.6260693\times10^{-34} J\cdot s$), k is the Boltzmann constant($1.3806505\times10^{-23} J\cdot K^{-1}$), c is the speed of light ($2.9979\times10^{8} ms^{-1}$), λ is the wavelength in meters, and T is the absolute temperature in degrees Kelvin.(Hudson, 1969)

(4) Stefan-Boltzmann Law. For a blackbody which is a perfect emitter, the total exitance M for the whole spectral range is obtained from the Stefan-Boltzmann law where M is defined as

$$M(T)=\sigma T^4. \quad (W\times m^{-2}) \qquad (2.4)$$

The parameter σ is the Stefan-Boltzmann constant equal to $5.670400\times10^{-8} W\cdot m^{-2}K^{-4}$ and T is the absolute temperature in degrees Kelvin over a wavelength range. (Hudson, 1969) However, most objects are grey bodies that have an emissivity independent of wavelength. Therefore, the exitance M of a grey body is defined as

$$M(T)=\varepsilon\sigma T^4, \quad (W\times m^{-2}) \qquad (2.5)$$

where ε is the emissivity, and $0<\varepsilon<1$.

b. Detectors and Electro-Optics System

The crucial part in the acquisition chain to obtain images from the scene is the detectors and electro-optics systems in the device. These systems represent the target properly only when they capture the necessary radiation or correctly depict the intensity of the radiation.

(1) Detectors. Detectors can be classified mainly into two types: photon and thermal detectors.

- Photon Detectors. Photoconductive, photovoltaic and photo-emissive devices are known as photon detectors. Photoconductive detectors are made up of semiconductor materials that can convert photons to internal charge carriers, which in turn causes the resistance of the material to change. This change in resistance reflects the rate of photons that is picked up. Photovoltaic (PV) detectors are constructed from a PN junction, and the voltage read out from the PN junction corresponds to the rate of photons incident on the detector. Lastly, photo-emissive detectors work differently from the rest of the photon detectors in that free electrons are emitted proportional to the photons being absorbed. Secondary devices are needed to capture and convert the electrons to electric signals.

- Thermal Detectors. Bolometers and pyroelectrics are classified as thermal detectors. A pyroelectric type is a combination of ferroelectric material and an electrode at each end. The ferroelectric material changes its polarization with temperature, and an external circuit measures the change in capacitance due to this change. Next, a bolometer is a device for measuring incident electromagnetic radiation through its resistance change; a microbolometer is scaled suitably for IR detection.

(2) Focal Plane Array. The radiation from the background scene is optically focused onto the detector focal plane array (FPA) of the detector through the use of optics. A focal plane array is a two dimensional array of detector elements simultaneously sampling separate portions of the scene. The scene field of view (FOV) is focused on an image plane by the optical system. The image is spatially sampled either in serial by scanning one or a few detectors over the image area or in parallel with a staring array of detector elements. A secondary setup comprising a charge coupled detector (CCD) may be used to read out the sampled signal in an orderly manner for reconstruction on the screen. Figure 5 below illustrates the read out mode of the focal plane array.

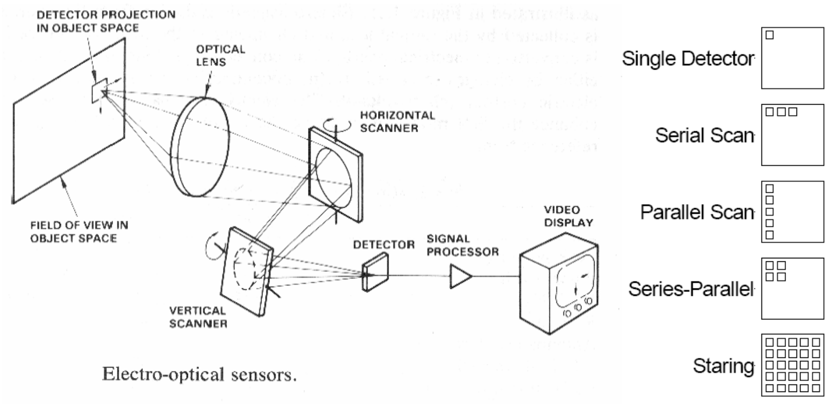


Figure 5. Read Out Mode of a Focal Plane Array (From: Graham, 2006)

For staring mode operation, the signal from the FPA detector, which normally covers the entire scene, is read out in a single cycle without having to scan through the scene. The radiation corresponds to a voltage output from the detector and the signal is read out to construct a pictorial representation of the scene through its aperture. Figure 6 below shows two FPAs designed in scanning and staring mode. Various methods exist to read out the signals from the detectors, however, they will not be discussed further, as they are outside the scope of this work. Specific details may be found in Driggers (1999).

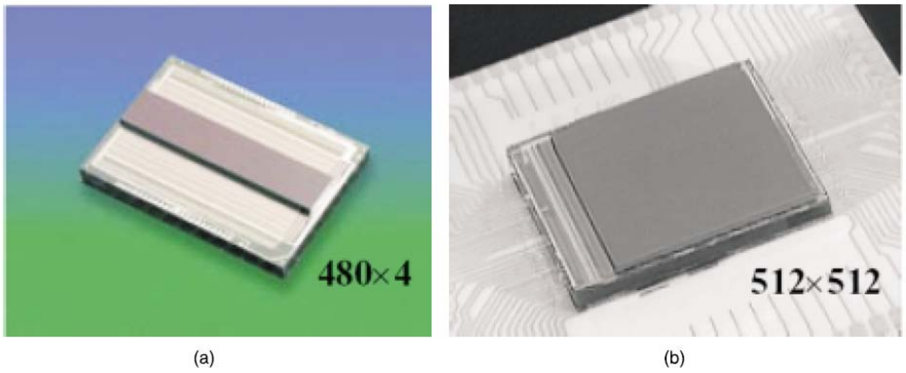


Figure 6. Second-generation PV HgCdTe Detectors, Which are Configured in (a) Parallel Scan with Time Delay Integration (TDI) and (b) Staring (parallel) Sensor Formats (From: Norton, 2002)

c. Performance Criteria

Many performance criteria are considered when designing and fabricating thermal imager detectors. These factors quantify how well a detector functions and the detection capability performance, and are discussed next.

(1) Responsivity. Firstly, the detectors' responsivity reflects the amplitude of the electrical signal relative to the incident flux. The responsivity R is a function of the incident flux wavelength and temporal modulation frequency and it is represented by

$$R(\lambda, f) = \frac{V_{out}}{\Phi_{incident}} = \frac{V_{out}}{E(\lambda, f)A_d}, \quad V/W \quad or \quad A/W \qquad (2.6)$$

where V_{out} is the detector output voltage, $\Phi_{incident}$ is the incident flux in watts, $E(\lambda, f)$ is the incidence irradiance in watts per square centimeter, A_d is the detector area in square centimeters and f is the fluctuation noise frequency (Driggers, 1999). The responsivity of the detector specifies how sensitive the device is to small changes in the scene temperature. The responsivity is also related to the ability of the detector to pick up targets that may be of interest but may not have drastic temperature differences from the environment. Thus, the higher the responsivity the more sensitive the device is to small temperature differences

(2) Noise Equivalent Power. The Noise Equivalent Power (NEP) quantity as defined by Brooker (2006) is the power input to the detector that results in a signal-to-noise ratio (SNR) equal to one. NEP is useful to determine the overall system SNR based on the noise equivalent flux on the detector. (Diggers, 1999) In the environment that the detector is operating in, there may be various types of noise that may mask the detector from obtaining the required signal. Possible noise types include

- kT Noise – Comprises of Johnson noise and thermal noise. Johnson noise is thermal noise generated from resistive material either from the detector itself or from external circuitry. Thermal detectors are affected by thermal noise. These noise sources exist within all materials that are above the absolute zero temperature at zero Kelvin. To overcome these thermal noise effects detectors are cooled to a lower operating temperature at 77 degrees Kelvin. The drawback for such devices is the additional cooling apparatus and the special enclosure needed for the detectors to function well, even though the resulting sensitivity may be much higher than that of other non-cooled thermal detectors such as a micro-bolometer.

- Shot noise occurs when the absorption of photons causes random generation of charge carriers. When recombination of charges also occurs, the noise generated in the process is called "generation-recombination" noise.

- 1/f noise is serious only at low frequencies, involves mechanisms only partially understood.

NEP is defined as the optical power that gives output SNR of 1, so that

$$NEP = \frac{I_{noise}}{R}, \quad Watt \qquad (2.7)$$

where the parameter I_{noise} is the noise current and R is the responsivity of the detector.

(3) Detectivity. Lastly, the detectivity (D) of the detector is reciprocal to NEP, which represents the sensitivity of the detector. Since low NEP represents high sensitivity, the quantity represents the detector's response

to low incident flux and the ability to detect low fluctuations. To better represent different types of detectors based on the material and noise frequency bandwidth of operation, a specific detectivity value known as “dee star” (D*) is derived. D* is defined as

$$D^* = \frac{\sqrt{A_d \Delta f}}{NEP}, \quad \frac{cm \times Hz^{\frac{1}{2}}}{Watt} \qquad (2.8)$$

where the parameter A_d is the detector area, Δf is the noise equivalent bandwidth of the system and high D* denotes high sensitivity. The D* value is regarded as a quality measurement of the detector sensitivity regardless of its size, wavelength of operation and material, since it relates to the SNR quantity which is the most universally used standard of measurement for equipment performance.

B. IMAGE FUSION

The basic idea behind Image fusion is to digitally combine two complementary images to obtain a better image that provides more information than the two images separated. Digital image enhancement has led to improvements in image quality via digital image manipulations. Several fusion algorithms and techniques have already been proposed in the image fusion community including “statistical and numerical” and “multi-resolution analysis” methods. Statistical and numerical method include the “Principal Component Analysis (PCA) and Principal Component Substitution” and the “Peli-Lim algorithm” proposed by Therrien, Scrofani, and Krebs (Therrien, Oct 1997). “Multi-resolution analysis” methods include “Pyramid-based Fusion” proposed by Burt and “Wavelet Transform Fusion” by Li, Manjunath, and Mitra. (Li, 1994)

This work focuses on wavelet-based fusion schemes, which are discussed next.

III. IMAGE FUSION METHODS

This chapter examines two wavelet-based fusion methods: pixel-based wavelet fusion schemes and region-based fusion which build on the watershed transform. This chapter provides a brief introduction to a reader without prior knowledge of the wavelet transform, wavelet transform fusion and region-based fusion approaches.

A. WAVELET TRANSFORMATION FUSION

Wavelet transform fusion involves decomposition of the original images into wavelet coefficients and combining them into a single image using a fusion scheme. The level of wavelet decomposition considered is selective and can be application-oriented. However this approach requires that the fusion step be done only at the same level of decomposition of each of the images. In the following sub-sections, the essentials of wavelet transform fusion such as wavelet transform, application of discrete wavelet transform (DWT) to image fusion, fusion at the decomposition level and the fusion schemes are examined. Figure 7 explains the overall wavelet transformation fusion process.

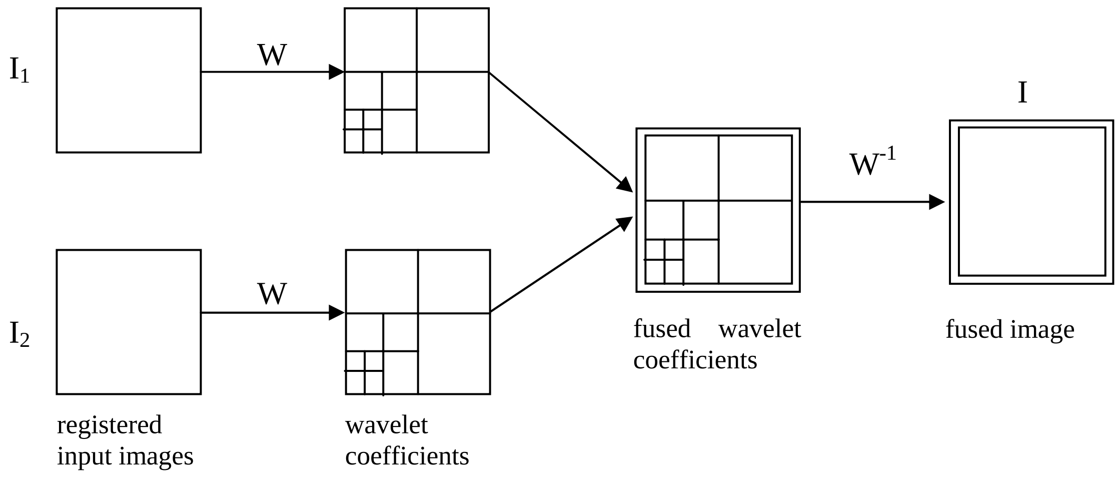


Figure 7. Fusion of the Wavelet Transforms of Two Images (From: Nikolov et al., 2002)

1. Wavelet Transform

Wavelet transformation is somewhat similar to the Fourier transform where raw signals can be broken down into frequency components to be represented in the frequency spectrum. What is missing in the Fourier transform is the temporal information when the signal is Fourier transformed to the frequency spectrum. Wavelet transformation evolved from the need to preserve the temporal information when representing the signal in the frequency spectrum. Wavelet transforms belong to the class of time-frequency transformations, which allow the user to know when specific frequency components occur in a signal. Different basis functions can be used for time-frequency representation. In wavelet transformations, two specific functions known as the mother wavelet and the scaling function are used to compute the wavelet transform.

Two main possible wavelet transform approaches are available to transform a signal into wavelets for analysis, continuous wavelet transform (CWT) and discrete wavelet transform (DWT). The CWT is computationally intensive and more information about this approach can be found at Polikar (1995). In this work we focus on the DWT as this transform has been shown to be significantly more computationally efficient.

2. Discrete Wavelet Transform: Implementation to Image Fusion

Filters with different cutoff frequencies were used to analyze the signal at different scales in the DWT approach (Polikar, 1995). This operation is accomplished by passing the signal through a series of high pass and low pass filters and filtering out the low frequencies for the analysis of the high frequencies and the high frequencies for the analysis of the low frequencies.

According to Nikolov et al., in one dimension the aim of the wavelet transform is to represent the signals as a linear combination of wavelets. Thus, using wavelet decomposition properties, a signal *f(t)* may be expressed as

$$f(t) = \sum_{m,n} c_{m,n} \psi_{m,n}(t), \qquad (3.1)$$

where $\psi_{m,n}(t)$ is the dilated and/or translated version of the mother wavelet given by the equation

$$\psi_{m,n}(t) = 2^{-m/2}\psi\left[2^{-m}t - n\right], \qquad (3.2)$$

where m and n are integers.

As mentioned previously, every level of decomposition yields approximate coefficients (obtained from low-pass filtering) and detail coefficients (obtained from high-pass filtering). The approximate coefficient of a function at resolution level 2^m is $a_{m,n}$ and at resolution level 2^{m-1} is $a_{m-1,n}$. The approximate coefficient expression is given by

$$a_{m,n} = \sum_k h_{2n-k} a_{m-1,k}, \qquad (3.3)$$

where h_n is the low pass finite impulse response (FIR) filter impulse response and $a_{0,n}$ is the sampled signal. $c_{m,n}$ represents the detail coefficients and is given by

$$c_{m,n} = \sum_k g_{2n-k} a_{m-1,k}, \qquad (3.4)$$

where g_n is the high pass finite impulse response (FIR) filter impulse response and $a_{0,n}$ is the sampled signal.

The one-dimensional wavelet transform can be extended to two dimensions. In such a case, filtering and down-sampling operations need to be carried out in horizontal and vertical dimensions. The resultant is an arrangement of four subbands expressed as horizontal frequency first followed by the vertical frequency second. These subbands are high-high (HH), high-low (HL), low-high (LH) and low-low (LL) images. (Nikolov et al., 2002) In this essence, the low-low subband represents the approximate coefficients. So by applying the same decomposition to the LL subband, a multi-resolution decomposition of the original image is feasible. The reverse is carried out for the reconstruction of the fused image, where up-sampling is applied instead of down-sampling and the complement of high pass and low pass FIR filters are used instead. Figure 8 illustrates the multi-resolution decomposition of an image.

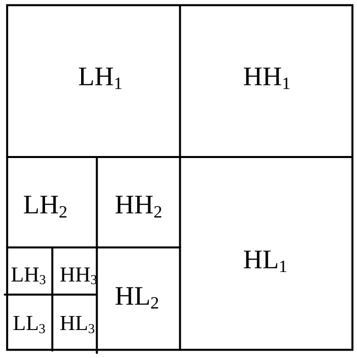


Figure 8. Subbands Representation in a Multi-resolution Decomposition of an Image

Figure 9 and Figure 10 illustrate a decomposition tree and reconstruction tree, respectively.

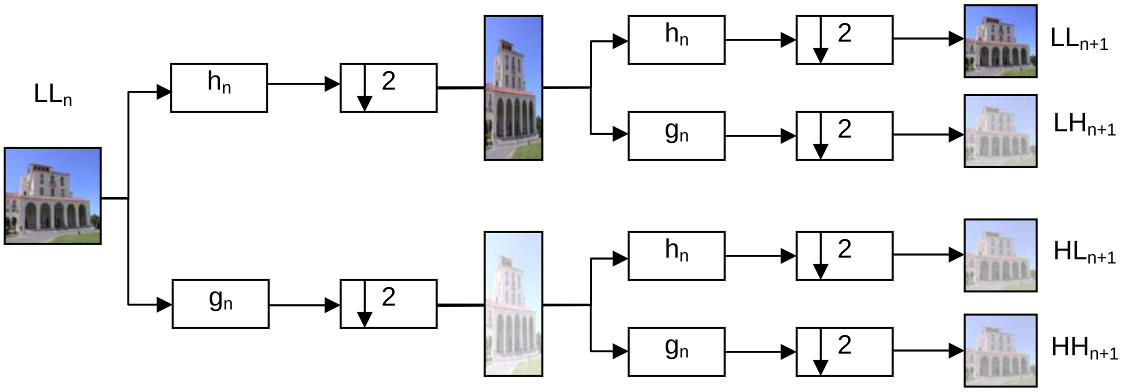


Figure 9. Decomposition Tree of the Original Images into Four Subband Images. (After: Chow 2004)

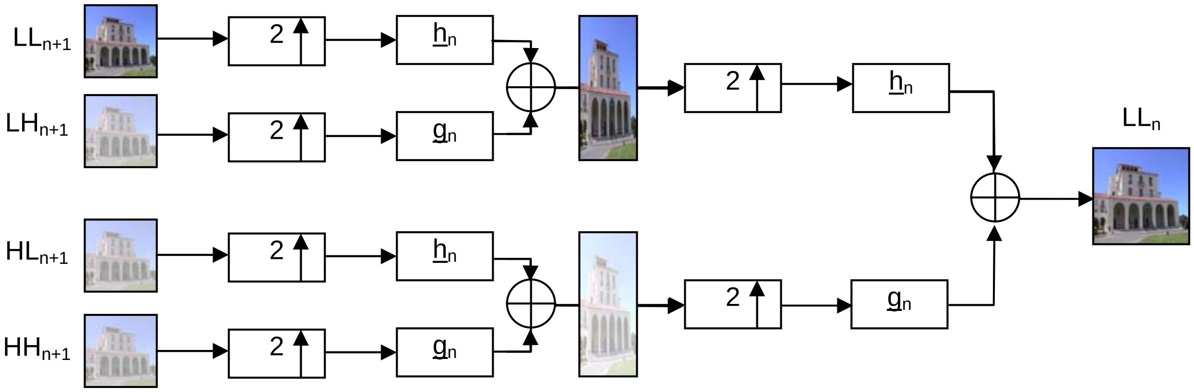


Figure 10. Reconstruction Tree from 4 Subband Images Back to Recover the Original Image. (After: Chow 2004)

The four subband images obtained after the decomposition phase relate closely to the original image. The approximate coefficients LL_{n+1} represent a coarse replica of the original without the details and the small variations. The coefficients LH_{n+1} represent the detail horizontal fluctuations from the original image and the coefficients HL_{n+1} represent the detail vertical fluctuations in the image. Finally, HH_{n+1} coefficients represent the diagonal fluctuations in the image which relate closely to the diagonal features in the image. The ability for subsequent approximate coefficients to be further broken down allows comparison of the two images' features in a multi-resolution approach. The multi-resolution also allows the individual coefficients from the images to be combined based on a fusing scheme and to carry out an inverse DWT to recover a fused image. Note the wavelet function or the mother wavelet plays an important role in the decomposition step. Many wavelet families have been developed over the years with different properties, and a brief introduction to the wavelet types used in this work is given next.

3. Wavelet Families

Wavelet decompositions are well suited to detect edges, which is especially important when the images considered have low contrast or the features in the images are not distinct. Even though pre-decomposition techniques such as image enhancement could be carried out to improve edge detection, it is important that the wavelet selected can perform to an acceptable standard even when there is prior image enhancement.

A study on the comparison of several wavelet families conducted by Chaganti (2005) suggested that Haar wavelets (Haar) have better performance for edge detection applications than Daubechies (db) and Coifman wavelet (Coiflet/Coif) families do when dealing with first level decompositions. Chaganti showed that Coiflet wavelets have better edge detection performances than db and Haar wavelet families when applying a second order level decomposition when noise is present in the image. This study also showed that the performance of Db and Coiflet are similar when dealing with third order level decompositions and the images under consideration are noisy, which is typical of night vision images. Chaganti also compared Biorthogonal (Bior), Haar wavelet decompositions, Db and Coiflet decompositions. He found out that Haar performance at first level is better than the rest of the wavelet families. However, Db and Coiflet perform better at higher level, with db more resistant to noise and Coiflets ability to pick up more details.

The focus of the thesis is to select a few commonly used wavelet families as bases before expanding to others. Since Db and Coiflet performances over a series of images were shown to be more consistent, we chose to work on the variations of Db and Coiflet. In addition, we also included, the symlet (Sym) family as it exhibits similar symmetry properties as Coiflets..

A similar experiment as conducted in Chaganti's study was applied to the selected wavelet families to test their suitability in the application considered. By observing the wavelet decomposition images, we were able to examine the edge detection ability of the wavelet families. Figure 11 shows the wavelet families mentioned.

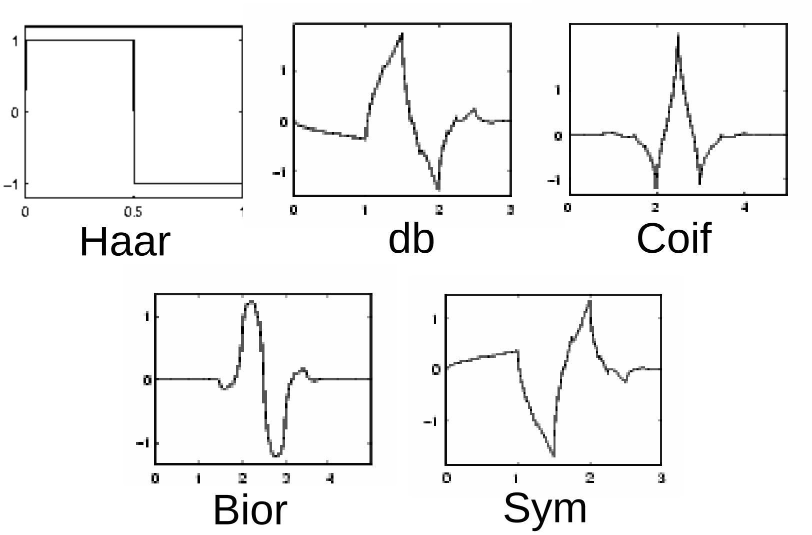
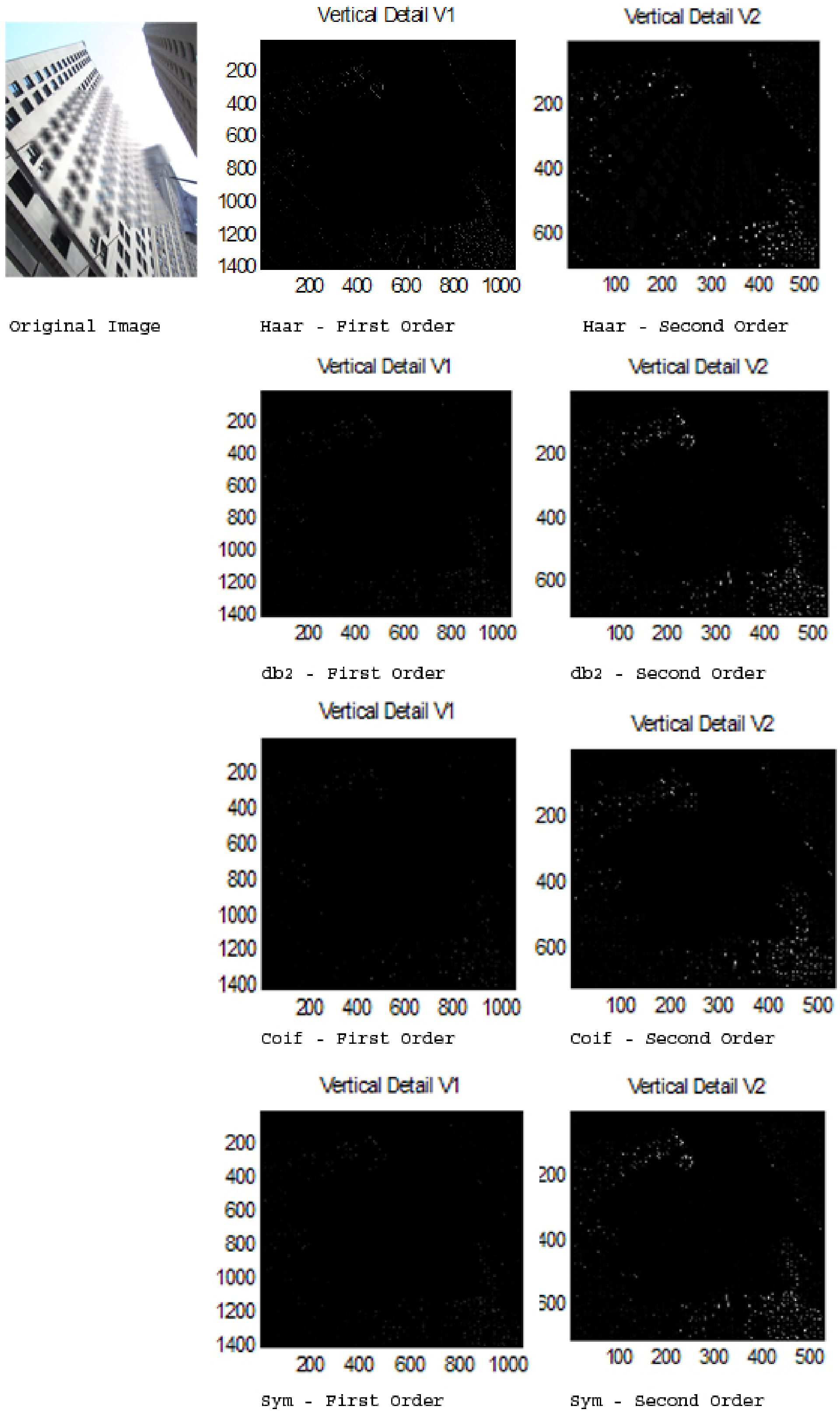


Figure 11. Wavelet Families (From: Mathworks, 2006)

The initial experiment was conducted for one set of pictures with some image processing done to illustrate the edge detection capabilities of the wavelet families. Similar to the results obtained by Chaganti (2005), Haar demonstrated better edge detection than db, Coiflet and Symlet at the first level of decomposition. However, when the second and third order decompositions were carried out, db, Coiflet and Symlet wavelets demonstrated more impressive performances than Haar. The higher orders of db, Coiflet and Symlet performed better than lower order ones. Figure 12 illustrates the results for the experiment using one of the images reduced to first and second order decomposition levels only.

Note that the picture used for this experiment has a lot of edges. This allows a better comparison and easier identification of unique patterns, such as a better detection capability for horizontal and vertical edges for certain wavelet families etc. Db performances were shown to be consistent over the various orders of decomposition considered. Symlet exhibited similar performances for second and above orders of decomposition. Since Symlet also have symmetry properties, it is ideal to include it as one of the wavelet families to be used for the fusion of images. Both Coiflet and Symlet wavelets were able to pick up more details than Haar and db were, likely due to their symmetry properties. Haar wavelets performance on the level 2 decomposition was coarse and results showed Db wavelets exhibited better performance than Haar wavelets in level 2 and above decomposition.

With these observations, db, Coiflet and Symlet families were selected to be used in the image fusion application.



Original Image

Haar - First Order

Haar - Second Order

db2 - First Order

db2 - Second Order

Coif - First Order

Coif - Second Order

Sym - First Order

Sym - Second Order

Figure 12. Experiment for Edge Detection of Wavelet Families.

4. Image Fusion

Fusion of images can be carried out at different levels of decomposition, depending on the type of methods and schemes adopted. The three levels of information representations are pixel, feature and decision level. Pixel level fusion takes place by comparing individual pixels that are spatially identical in two separate images and fusing them using a fusion scheme. This method assumes that the images are registered prior to fusion, and image resampling and image registration operations are needed prior to fusion when this is not the case.

a. Image Resampling

Images to be fused must have the same size in order for wavelet transform fusion to take place, and images have to be resampled using an interpolation technique when there have different sizes.

b. Image Registration

For a proper fusion of two separate images, both images have to be accurately aligned and registered prior to merging the images. This property is especially important when the images include numerous edges

For this thesis, the images are assumed to be registered and the image resampling and registration will not be discussed further. Feature and decision level fusion look at larger areas within the image for decision on how the images would be fused. The wavelet transform fusion method is a form of pixel level fusion if surrounding pixels are not considered in the fusion scheme. Wavelet transform fusion may evolve into feature level or decision level fusion when further processing of the image is done, or extraction of information pertinent to the image is done at its decomposition level. Processing and information extraction such as segmentation into regions, characterization into shape, size, contrast texture and intensity from the image means that the fusion method logically develops into the feature level. The fusion method is categorized as decision level fusion when other inputs such as the amount in which individual images contribute to the fused image are taken into account. Extending from the pixel level of wavelet transform fusion is the region-based fusion which incorporates region segmentation using the watershed transform and region based fusion schemes. The region-based fusion approach can be categorized as a feature-level fusion scheme and will be covered later on in the chapter. For the wavelet transform fusion step to be complete, fusion schemes have to be applied before an inverse wavelet transform is carried out. The fusion scheme decides how the coefficients at the decomposition level are to be combined. Fusion schemes can range from a simple summation of coefficients to more complicated methods of sampling the surrounding pixels of the pixel of interest and determining its contribution amount to the fused image. Next we describe the fusion schemes considered in this study.

5. Fusion Schemes

The three fusion schemes described in this section were adopted in the earlier work of Chow (2004) which selected approaches commonly used in the image fusion community. The general framework is shown in Figure 13 using registered images.

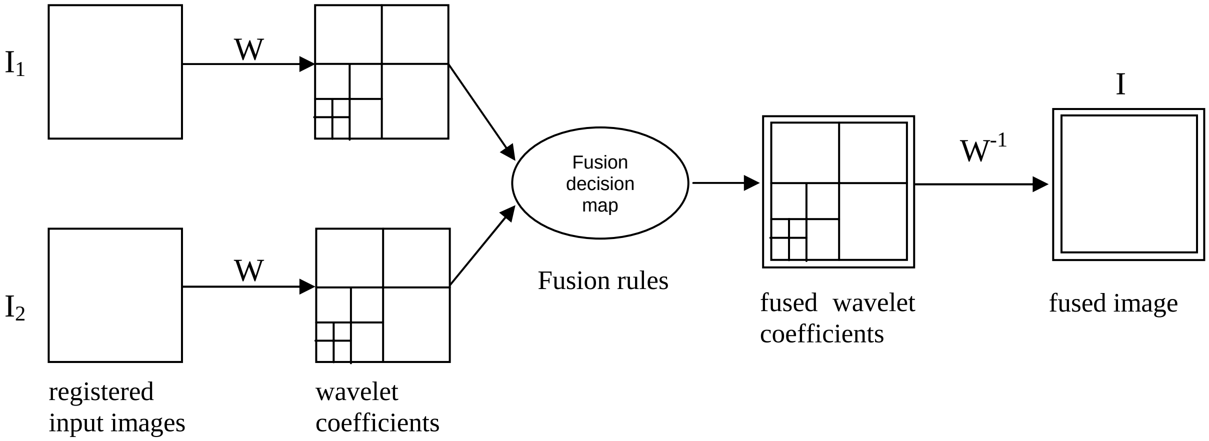


Figure 13. General framework for wavelet transforms fusion. (After Chow, 2004)

Details coefficients obtained after the wavelet decomposition step represent the image salient features. Next, detail coefficients are compared one pixel at a time to determine their level of contribution to the resulting fused image they are. A fusion rule governs the decision and determines how the resultant detail coefficients highlight the salient features in the fused image; this step is known as the fusion decision map. Approximate coefficients are processed differently. Approximate coefficients obtained from the two images are averaged together to form the new approximate coefficients to the fused image. Figure 14 illustrates the framework for the formation of the fusion decision map.

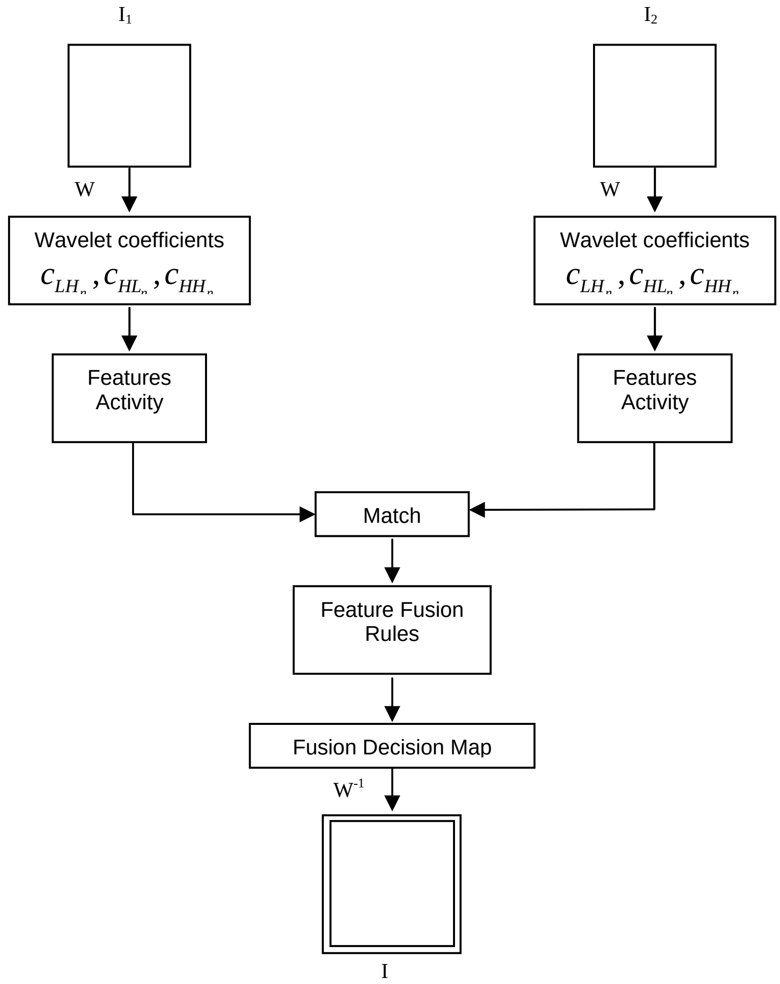


Figure 14. Framework for the Formation of the Fusion Decision Map. (After Chow, 2004)

Pixel activity level, i.e., its salience in the image, is measured using a very simple method; a pixel is said to be salient when its wavelet coefficient associated to that location has high value. Thus, the saliency of the coefficients is determined by using the coefficients absolute values as:

$$a_{A_n}(j,k) = \left|c_{A_n}(j,k)\right| \quad and \quad a_{B_n}(j,k) = \left|c_{B_n}(j,k)\right|, \qquad (3.5)$$

where the parameters $c_{A_n}(j,k)$ and $c_{B_n}(j,k)$ represent the nth level wavelet coefficients at location (j,k) of input image A and B, respectively (Chow, December 2004).

If we expand the area from the pixel of interest to surrounding pixels, one may have a more sophisticated possibility of getting a salient feature from the image. The method known as window-based activity measure (Chow, 2004), allows a selective window size for the operation. A typical 3-by-3 or 5-by-5 window size is chosen because even though having a bigger window size means a more robust fusion system, it also results in heavier computational load. The activity associated with the nth level pixel centered in the window at location (j,k) using a window-based approach is determined using

$$a_{W(A,B_n)}(j,k) = \sum_{s\in S,t\in T} c_{A,B_n}(j+s,k+t), \qquad (3.6)$$

where the variable $a_{W(A,B_n)}(j,k)$ is the window-based activity measure and the parameter S and T are sets of horizontal and vertical indexes that describe the current window (Chow, 2004).

Next, a matching criterion is used to have a higher degree of accuracy when comparing the two images. If the two pixels have the same intensity or are closely related to each other, this information is used to decide how the pixel is treated. The matching criteria $m_{F_n}(j,k)$ defined in the range 0 to 1 represents the correlation between the corresponding pixels at location (j,k) for the nth level coefficients and is defined as (Chow, 2004)

$$m_{F_n}(j,k) = \frac{2c_{A_n}(j,k)c_{B_n}(j,k)}{\left|c_{A_n}(j,k)\right|^2 + \left|c_{B_n}(j,k)\right|^2}. \qquad (3.6)$$

Next, fusion rules are developed to allow the decision on the type of treatment to the individual pixels to be carried out. Three fusion rules discussed earlier by Chow (2004) are adopted again in this work; two pixel-based and one window-based scheme. Further details regarding these schemes may be found in Chow (2004).

a. Fusion Rule 1 – Selection of Dominant Mode

For each pixel in the image, this fusion rule selects the most dominant feature of the two images pixels, i.e., the pixel with the highest activity level. Thus, the fusion rule is defined as:

$$c_{F_n}(j,k) = \begin{cases} c_{A_n}(j,k) \ if \ \left|a_{A_n}(j,k)\right| > \left|a_{B_n}(j,k)\right| \\ c_{B_n}(j,k) \ if \ \left|a_{B_n}(j,k)\right| > \left|a_{A_n}(j,k)\right| \\ \dfrac{c_{A_n}(j,k)+c_{B_n}(j,k)}{2} \ if \ \left|a_{A_n}(j,k)\right| = \left|a_{B_n}(j,k)\right| \end{cases}, \qquad (3.7)$$

where the variable $c_{F_n}(j,k)$ is the composite coefficient from the two images' detail coefficients, $a_{A_n}(j,k)$ and $a_{B_n}(j,k)$ are the absolute values of the coefficients of image A and image B respectively.

b. Fusion Rule 2 – Weighted Average of Modes

This approach is based on a weighted combination of the source images. The matching measure defined previously is used to determine the respective contribution by both images and is given by

$$c_{F_n}(j,k) = \begin{cases} wc_{A_n}(j,k)+(1-w)c_{B_n}(j,k) if \ \left|a_{A_n}(j,k)\right| > \left|a_{B_n}(j,k)\right| \& m_{F_n}(j,k) \le T \\ wc_{B_n}(j,k)+(1-w)c_{A_n}(j,k) if \ \left|a_{B_n}(j,k)\right| > \left|a_{A_n}(j,k)\right| \& m_{F_n}(j,k) \le T \\ \dfrac{c_{A_n}(j,k)+c_{B_n}(j,k)}{2} \ if \ m_{F_n}(j,k) \ge T \end{cases}, \qquad (3.8)$$

where w is the weighted value that defines the contribution of the selected coefficient, $m_{F_n}(j,k)$ is the matching value and T is the pre-defined threshold. A higher weight is given to the pixel with a higher activity level and T is set at 0.8, which means an 80% match for the pixel of both images. At the present moment, the threshold value$T$ was set at 0.8 by trial and error for the application considered however it could be adjusted if further analysis showed a more suitable value exists.

c. Fusion Rule 3 – Weighted Average of Window-based Modes

Expanding on Rule 2, neighboring pixels are taken into account when determining the weight of each corresponding pixel from both images. From the window-based activity measure defined previously, the measure is incorporated into the fusion rule as

$$c_{F_n}(j,k) = \begin{cases} wc_{A_n}(j,k) + (1-w)c_{B_n}(j,k)\, if\ \left|a_{W(A_n)}(j,k)\right| > \left|a_{W(B_n)}(j,k)\right| \& \, m_{F_n}(j,k) \le T \\ wc_{B_n}(j,k) + (1-w)c_{A_n}(j,k)\, if\ \left|a_{W(B_n)}(j,k)\right| > \left|a_{W(A_n)}(j,k)\right| \& \, m_{F_n}(j,k) \le T\,, \\ \dfrac{c_{A_n}(j,k) + c_{B_n}(j,k)}{2}\ if\ \ m_{F_n}(j,k) \ge T \end{cases} \tag{3.8}$$

where the variables $a_{W(A_n)}(j,k)$ and $a_{W(B_n)}(j,k)$ are the window-based activity level measures for image A and B.

B. REGION- BASED FUSION

An image can usually be segmented according to the objects and features combined within, which also applies to low contrast images such as night vision and thermal images. The application of segmentation is easier on night vision and thermal images as the high intensity regions define either area of light sources or heat sources. The composition of these images will be explored in the next chapter. The region-based fusion may be considered as a more robust method and is able to better interpret the feature intensity in the images than pixel-based schemes are. By including the feature intensity in the decision of the fusion, important features may be better preserved and integrated in the fused image. As stated by Chow, the purpose of the region fusion is to optimally extract the information from different sources and maximize the “scene content” in the fused image.

1. Region Segmentation

As mentioned previously, an image may be segmented according to its objects and features. The author implements a region-based segmentation scheme on objects and features by considering the level of homogeneity in these regions, their strong statistical correlation and visual similarities.

2. Watershed Transform

The basic idea behind the watershed transform can be explained by drawing a parallel with natural features such as ridges and valleys that make up the geographical landscape of the earth. So an image may be segmented into high and low intensity regions by viewing it as a topographical map, where high intensity regions correspond to peaks and low intensity areas as valleys. The ridges forms watershed lines that channel water to form pools and reservoirs known as catchment basins at low lying regions. If this water has nowhere to flow, the water level will continue to rise until it overflows to another region possibility out of the catchment basin. The watershed transformation is a morphological transformation used to segment an image into regions of interest. As previously described if water starts to flood from the watershed into the valley, the water will rise uniformly throughout the low regions. If a dam is erected between the two regions where the flooded regions meet and subsequently for all regions, the image is partitioned according to different regions of regional minima. Once the catchment basin is created, the gradient image is formed using “Dilation” and “Erosion” morphological operations, which are discussed in detail in Chow (2004). Note that a direct application of the watershed transform to a gradient image usually produces excessive segmentation. Thus, the so called “maker-based segmentation” approach is usually preferred, which limits the over-segmentation problem encountered in the basic watershed transform approach. (Chow, 2004).The watershed algorithm is applied to the approximate coefficients of the wavelet transformed image rather than to the grayscale image to significantly reduce over-segmentation and remove broken contours, as originally proposed by Jung et. al. (2002). Chow found that generally a good segmented image only needs about forty or fewer regions after numerous experiments. This constraint also helped to ensure that the processing times for subsequent stages were curbed. Figure 15 and 16 illustrate the region segmentation process obtained using the approximate coefficients at various level of decomposition.

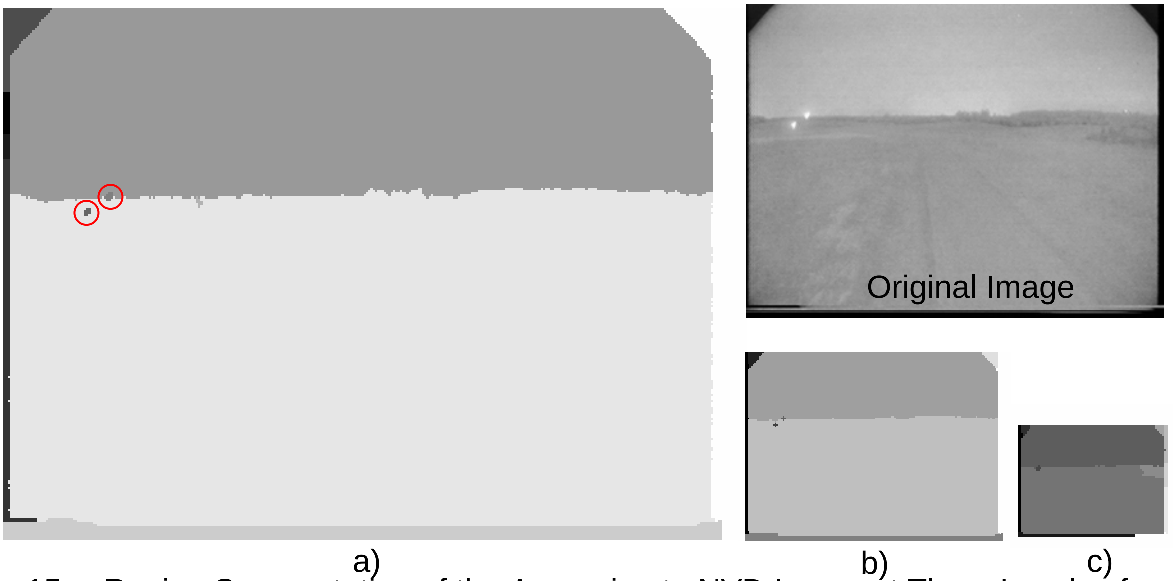


Figure 15. Region Segmentation of the Approximate NVD Image at Three Levels of Decomposition: a) Level 1, b) Level 2 and c) Level 3. (After: Chow, 2004)

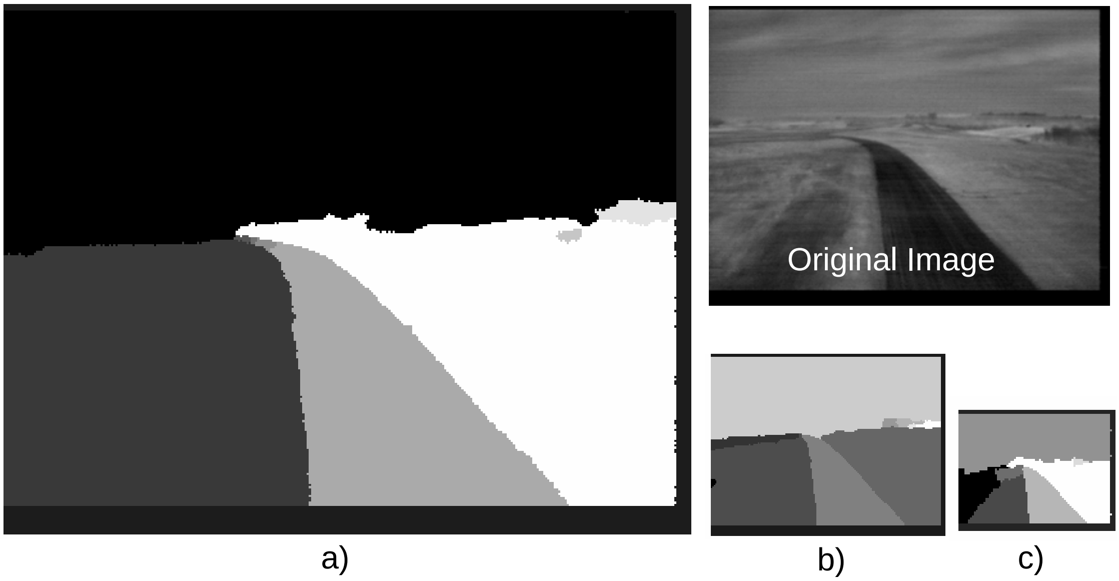


Figure 16. Region Segmentation of the Approximate Thermal IR Image at Three Levels of Decomposition: a) Level 1, b) Level 2 and c) Level 3. (After: Chow, 2004)

Thus, we limited the number of segmented regions to 40 in the fusion application, by adjusting the threshold value of the watershed transform. The experiment on narrowing down to the ideal threshold value and algorithm for adjusting the threshold will be explained later in Chapter V.

3. Fusion Methodology

After the images to be fused are segmented, regions unique to each image need to be merged at each level of decomposition to direct the fusion of the wavelet coefficients at each level. The fusion methodology is an extension of the wavelet transform framework with an addition of region activity and expands the "Feature Fusion Rules" to "Region and Feature Fusion Rules". As the name implies, the region information is included in the fusion decision map. The wavelet transform framework with the added features is shown in Figure 17.

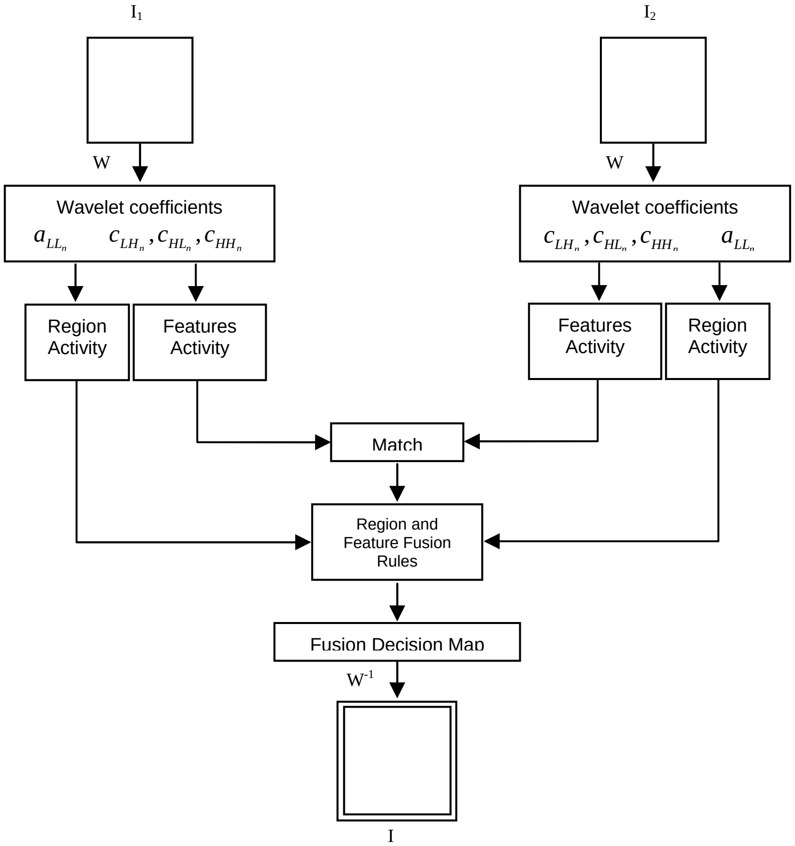


Figure 17. Framework for the Formation of the Fusion Decision Map for Region-based Fusion. (After: Chow, 2004)

a. Segmentation Process and Feature Activity Information

Each image generates a region activity map which comprised the segmentation of regions in the image. Two region activity maps are combined to form the joint region activity map. The joint region activity map is used in the fusion process together with the feature activity information to generate the fusion map. The feature activity information is explained later in this section. Figure 18 shows the construction of the joint region map.

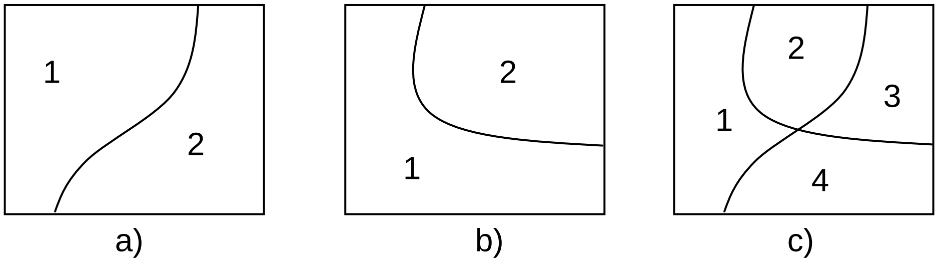


Figure 18. Region Segmentation: a) Region Representation of Image A; b) Region Representation of Image B and c) Joint Region Map, Indicating the Four Identified Regions. (From: Chow 2004)

The fusion rules are crucial in deciding how the regions in the images are treated. Using the joint region map, the regions are labeled using the following notation

$$R = \{R^k{}_n\}, \qquad (3.9)$$

where the variable $R^k{}_n$ represents the kth segmentation at level n. Following the definition of the joint region map, the sizes of the regions are determined. By overlaying the joint map over the original images, one is able to see the region segmentation superimposed on to the images.

The activity levels in each of the regions for image A, known as the feature activity information are computed using

$$A_{A_n}(x) = \frac{1}{S_i} \sum_{S_i} a_{A_n}(j,k), \qquad (3.10)$$

where $a_{A_n}(j,k)$ is the absolute value of the detail coefficient and represent the nth level activity measure at location (j,k) and S_j is the size of the region determined previously. A similar operation is carried out for image B. More details are available in Chow (2004).

b. Fusion Rule

Once the computations of the activity levels in each of the images are completed, the fusion can be carried out using specific fusion rules for region-based fusion. The fusion rule implemented here is a simple weighted average for the detail coefficients using

$$c_{F_n}(j,k) = \begin{cases} wc_{A_n}(j,k) + (1-w)c_{B_n}(j,k) \, if \, \left|A_{A_n}(x)\right| > T \\ wc_{B_n}(j,k) + (1-w)c_{A_n}(j,k) \, if \, \left|A_{B_n}(x)\right| > T \, , \\ \dfrac{c_{A_n}(j,k) + c_{B_n}(j,k)}{2} \, otherwise \end{cases} \qquad (3.11)$$

where the variable $c_{F_n}(j,k)$ is the composite coefficients from the two images' detail coefficients, w is the weighted value that defines the contribution of the selected coefficient, $A_{A_n}(x)$ and $A_{B_n}(x)$ are the activity levels in the region which the coefficients lies in image A and B, and T is the pre-defined threshold. The fusion rule selects the coefficients according to the weight assigned after considering the activity level in the region where the coefficients exist. If however, both coefficients exhibit similar activity levels, the average is taken and no weights are assigned. The ideal weight factor after experimenting with the fusion rule is set at 0.8. The threshold is similarly set at 0.8 to define a region of high activity.

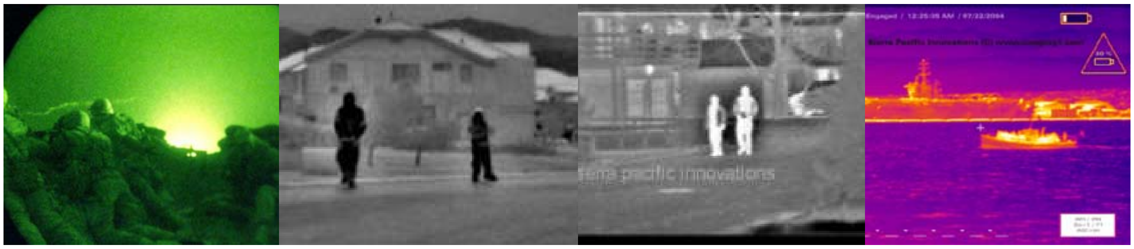
Chow noted in his study that this fusion rule retains the most important features to extract from both night vision and thermal images, which should help in better orientation and improved situational awareness. However, our results showed that the region-based image fusion does not necessary produces the best fused image. The segmentation in the region-based fusion may make the image look rather patchy, which may not appeal to most observers. On the other hand, the observer may perceive the image as being easier to interpret when the fused image maintains a certain level of uniformity.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. DIGITAL IMAGE PROCESSING

A. IMAGE CHARACTERISTICS

Night vision images are easily differentiable from thermal images by the ‘greenish’ glow which is absent from thermal images. Under low light conditions, the visual ability of the human eyes decreases. Since human vision is most sensitive to green and shades of green than other colors, NVD are designed with green phosphor screen. Thermal images on the other hand are displayed simply in ‘white’ hot or ‘black’ hot in most military systems. Commercially available thermal imagers use “False Color” imagery to display their thermal differences. These displays may be shown in grayscale, black-white or green monochrome screens. Figure 19 shows the types of night vision and thermal images in night vision and thermal devices currently available in the market.



(a) (b) (c) (d)

Figure 19. Night Vision and Thermal Images Examples (a) Night vision (From: Military, 2006), (b) Black Hot, (c) White Hot and (d) Thermo-flash Color Vision. (From: imaging1, 2006)

1. Composition

In order to adopt a suitable fusion scheme for night vision and thermal images, one needs to first understand the composition of the images that the fusion scheme needs to handle. First, the author will examine the individual composition of the night vision and thermal images. Next, we discuss how these image compositions are affected by atmospheric effects and finally propose the method to improve the image quality.

a. Night Vision

A typical third generation night vision device has the ability to boost the amplification level by eighty thousand times. Such amplification level in turn enables targets within a distance of 360 meters to be clearly detected under limited starlight (ATN Corporation, 2006). In addition, identification of closer targets is also greatly improved. Night vision images have a unique composition. We will provide an analysis of night vision images' composition by highlighting some of their distinct features.

(1) Foreground and Background. An image can often be separate in two distinct areas; the foreground and the background. For a night vision image, objects in the foreground are normally clear enough to be easily distinguishable. For example, targets such as soldiers may easily be differentiated to be either friends or foe based on the shape their uniforms and/or the equipment and weaponry that they operate. Due to the lack of color differentiation and atmospheric effects such as dispersion and scattering, the features at the background of most night vision images tends to be rather obscure and cluttered, making it difficult for the user to identify targets far away. It may still be possible to detect targets, but it becomes impossible to differentiate between friends or foes. Figure 20 below illustrates foreground and background issues in a night vision image.

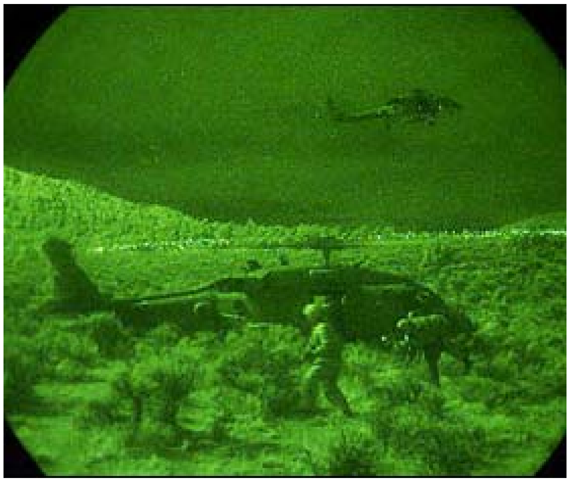


Figure 20. Soldiers at Foreground Clearly Identifiable, Light Sources and Helicopter at Background not as Differentiable.(From: Military. 2006)

(2) Light Sources. Local bright light sources in a night vision image may cause flooding of the image, with features such as halos and flare of light. The effect here is a result of the detector plane in which the diffusion of the electron image spreads the light image. Targets near light sources lose detail due to the flare effect from the light sources which causes the region to look 'hazy', as illustrated in Figure 21 shown below. This characteristic present in night vision images also reduces the ability for the user to identify details and outlines in the image. The halos that are created by the light sources also have the same effect on the image as the flare of light.

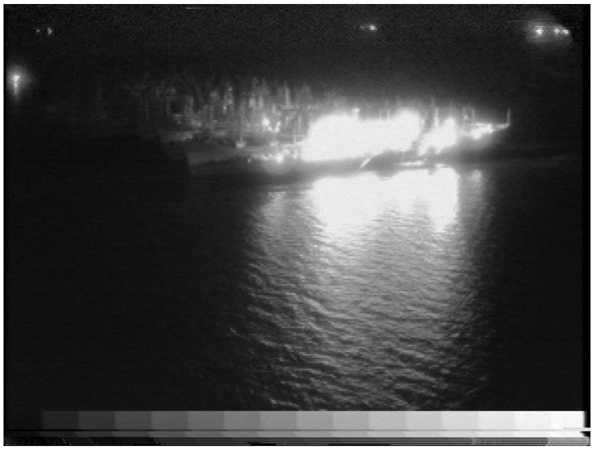


Figure 21. Flare Effect of Bright Lights on the Ships (Photograph courtesy of Naval Research Laboratory)

(3) Lack of Contrast. Due mainly to the flare effect of the ambient lighting which will be discussed in the section following this, night vision images tend to present lack of contrast in its images. Even when a night vision image exhibits greater ambient lighting and includes light sources, the image may have lack of contrast to its overall image. This reduces the ability for the targets or important features to standout from the surroundings or its environment. Figure 22 below exhibits this effect. (a) Shows the lack of contrast throughout the entire image even though there are light sources around and a relatively bright ambience, but the image still looks dark. (b) Has a brighter ambience probably due to moonlight effect, still lacking in contrast even when the image looks bright. We observe that the features at the background are not obvious.

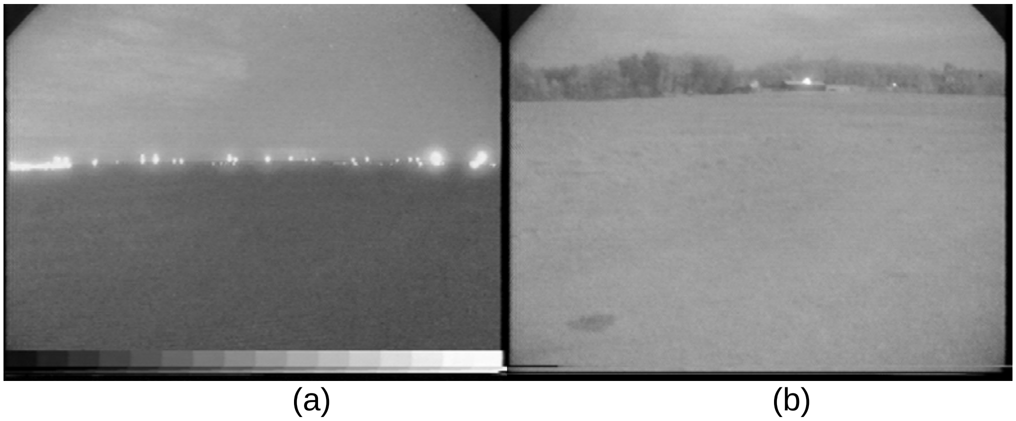


Figure 22. (a) Exhibiting an Overall Darkness and (b) Exhibiting an Overall Brightness. (Photographs courtesy of Naval Research Laboratory)

b. Thermal Image

Forward-looking infrared (FLIR) imagers, (a common term for thermal imagers) capture traditional thermal images that are simple to interpret and usually displayed in two formats with one type displaying hot objects as white and the other hot objects as black. Furthermore, current high-tech imaging devices can display color images and include a temperature scale along side the image to provide information on the exact temperature of the objects in the scene. Additional features also include zooming to the point of interest and the ability to output the video display to an external capturing device or to on-board recording capabilities. Figure 23 illustrates these possible features.

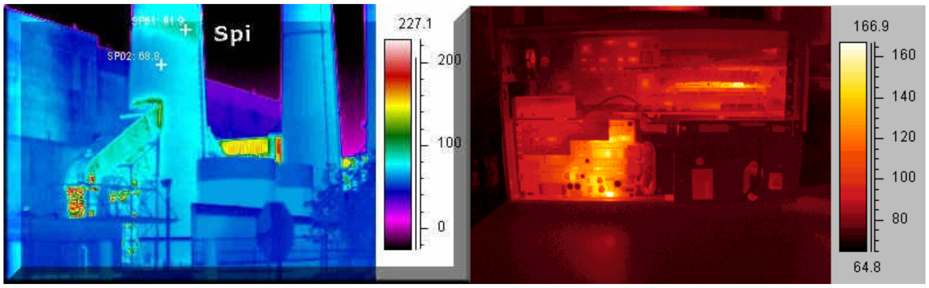


Figure 23. Examples of Color Display of Thermal Images. (From: Infrared1, 2006)

(1) Suppressed Background. One distinct advantage of TI over NVD is the ability to emphasize the hot bodies from their surroundings. The background is suppressed from the target, making it easier for the user to identify features in the thermal image. The ability for the thermal imager to do so

depends primarily on the scene acquisition process. When the detector captures the scene through the aperture of the device, in the subsequent process of obtaining an electrical signal output from the detector, ac coupling is used to remove signals below a certain threshold. This process results in having hot bodies to appear more intense than their background. However, the ac-coupling process results in loss of details in the IR image.

(2) Contrast Limited. Unlike a night vision image which provides information from light emission (both self-emitting and reflected) and near IR radiation, a thermal imager relies primarily on objects self-thermal radiation. As a result, a thermal image may present a lack of contrast in areas of high temperature homogeneity even though there may be slight temperature differences, whereas a night vision device may be able to highlight slight contrast differences in areas where the objects are similar. Similar observations were highlighted by Teo (2003).

(3) Similarity with Night Vision Images. Thermal images do demonstrate some similarities with night vision images. As night vision and thermal images are regarded as complementary forms of imaging to each other, each is able to set apart objects that the other may not be able to distinguish. The table below examines the features typical of a military setting that each system is able to identify. The tabulation is based on the available pictures for this thesis, as well as further exploration of those available on the internet. This assessment may be subjective as it is based solely on the opinion of the writer, however it will be useful in evaluating the improvement of the fusion process as the fused image is compared with the original images.

SYSTEMS	LIGHT SOURCES	HEAT SOURCES	SOLDIERS AND	VEHICLES (INCLUDIN G SHIPS)	BUILDINGS	WATER FEATURES	OPEN AREA	TRACKS AND	TREES AND	TERRAIN	CLOUDS
NIGHT VISION	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
THERMAL IMAGE		✓	✓	✓	✓	✓	✓	✓			✓

Table 2. Comparison of Night Vision and Thermal Images.

As highlighted by Teo (2003), TI imaging exhibits some advantages over the NVD imaging in its ability to perform under non-ideal conditions such as fog, smoke, rain and snow. The ability for longer wavelength IR to propagate better in atmospheric conditions than shorter wavelength of visible light and near-IR radiation as well as other effects of atmosphere is discussed in the next section.

2. Atmospheric Effects on Image Acquisition

Thermal radiation propagates better than light photons under certain atmospheric conditions partly as a result of absorption. It is able to propagate further as it is not severely affected by absorption. As light photons' wavelength matches the molecular resonances of the elements in the air, its attenuation is more severe. Night vision products work primarily in the visible and near-IR ranges, which have lower atmospheric transmittance (about 60 to 70% and a narrower window of transmittance). Another atmospheric effect affecting night vision devices is scattering.

(1) Atmospheric Absorption. Atmospheric characteristics, such as absorption for example, affect radiation propagation. Absorption occurs when the radiation wavelength matches the resonance frequency of the molecules present in the atmosphere such as oxygen (O_2), carbon dioxide (CO_2), nitrous oxide (N_2O), water (H_2O) and ozone (O_3), thus forming what is referred to as "transmittance windows". As large amount of these molecules are present in the atmosphere, they have a significant effect on the EM energy propagating through the air. The process of atmospheric absorption converts the electromagnetic energy to another form and is irreversible. Figure 24 depicts the atmospheric transmittance versus wavelength including the type of molecules that causes attenuation at specific wavelengths.

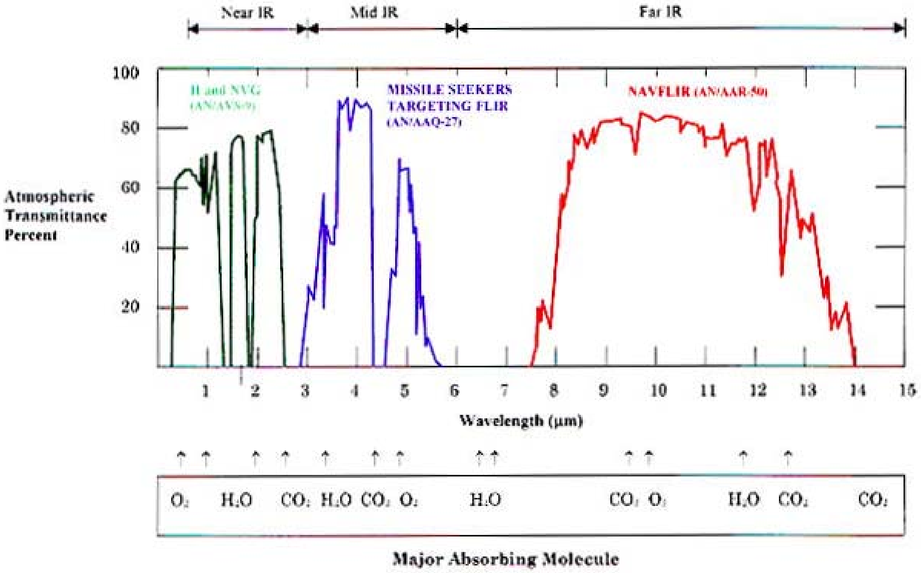


Figure 24. The atmospheric transmittance of electromagnetic radiation (From: Jane's August 2005)

(2) Scattering. Another natural phenomenon that affects EM propagation in the atmosphere is the effects of molecular aerosol scattering. The scattering of radiation by air molecules falls within the Rayleigh scattering regime and the scattering by aerosols in the air is called Mie scattering.

Guyot (1997) mentioned that Rayleigh scattering is due to the gas molecules of much smaller dimensions than the wavelengths of solar radiation. In this case, the quantity of light diffused is inversely proportional to the fourth power of a wavelength. Consequently, this scattering has an important role only where short wavelengths are concerned (λ<0.55μm). For example, it is for this reason that the sky is blue and that the setting sun appears red, because the short wavelengths are strongly scattered.

Also mentioned by Guyot (1997) is that Mie scattering is due to aerosols suspended in the air (dust and microdroplets of water with diameters between 10^{-2} and 10^{2} times that of the wavelength). Mie scattering is strongly dependent on ratio of particle size to wavelength and is present across the whole solar spectrum, but very dependent on the local aerosol conditions. The figures below depict the atmospheric effects from absorption and scattering relative to the wavelength of the EM radiation.

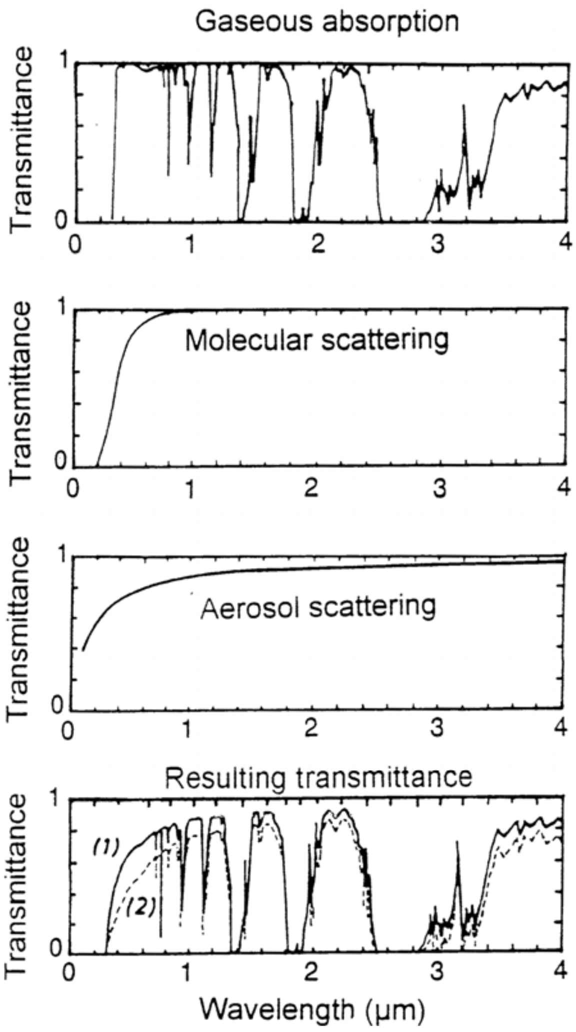


Figure 25. Atmospheric Effects : Absorption and Scattering on EM radiation (From: Guyot, 1997)

B. IMAGE ENHANCEMENT

For an object to be distinguishable in a scene, it needs to have a visual difference from its surroundings. This visual difference is a combination of color, brightness and texture difference known as a contrast, which is crucial in identifying objects in a scene. The higher the contrast difference between an object and its surrounding, the more distinguishable the object is. As explained in Wikipedia (22 November 2006), “In visual perception of the real world, contrast is determined by the difference in the color and brightness of the object and other objects within the same field of view. Because the human visual system is more sensitive to contrast than absolute luminance, we can perceive the world similarly regardless of the huge changes in illumination over the day or from place to place." In other words, increasing object contrasts make them more distinguishable from their surroundings. Taking note that night vision and thermal image inherently have low contrast, image quality may be improved by enhancing their contrast. However, this process is not without limitation as over-enhancing the contrast makes images appear cluttered.

Teo Chek Koon (thesis 2003) showed that contrast in night vision images could be significantly enhanced digitally by an application of the contrast limited adaptive histogram equalization (CLAHE) algorithm as a preliminary to fusion with corresponding thermal images. His research was used as the foundation to determine the type of digital image enhancement needed prior or after image fusion in this thesis. Teo focused on using the CLAHE approach to perform contrast equalization and enhance details in local areas of the image.

1. Image Histogram

Adaptive histogram equalization uses the image histogram to perform image enhancement. An image histogram is a plot of the distribution of intensities or gray levels found in the image. The histogram of a digital image with gray levels in the range [0, L-1] can be represented by the discrete function

$$p(r_k) = \frac{n_k}{n}, \qquad (4.1)$$

where r_k is the k^{th} gray level, n_k is the number of pixels in the image within that gray level, n is the total number of pixels in the image, and $k = 0,1,2,...L-1$.(Teo, 2003). Figure 26 illustrates typical variations found in digital image histograms.

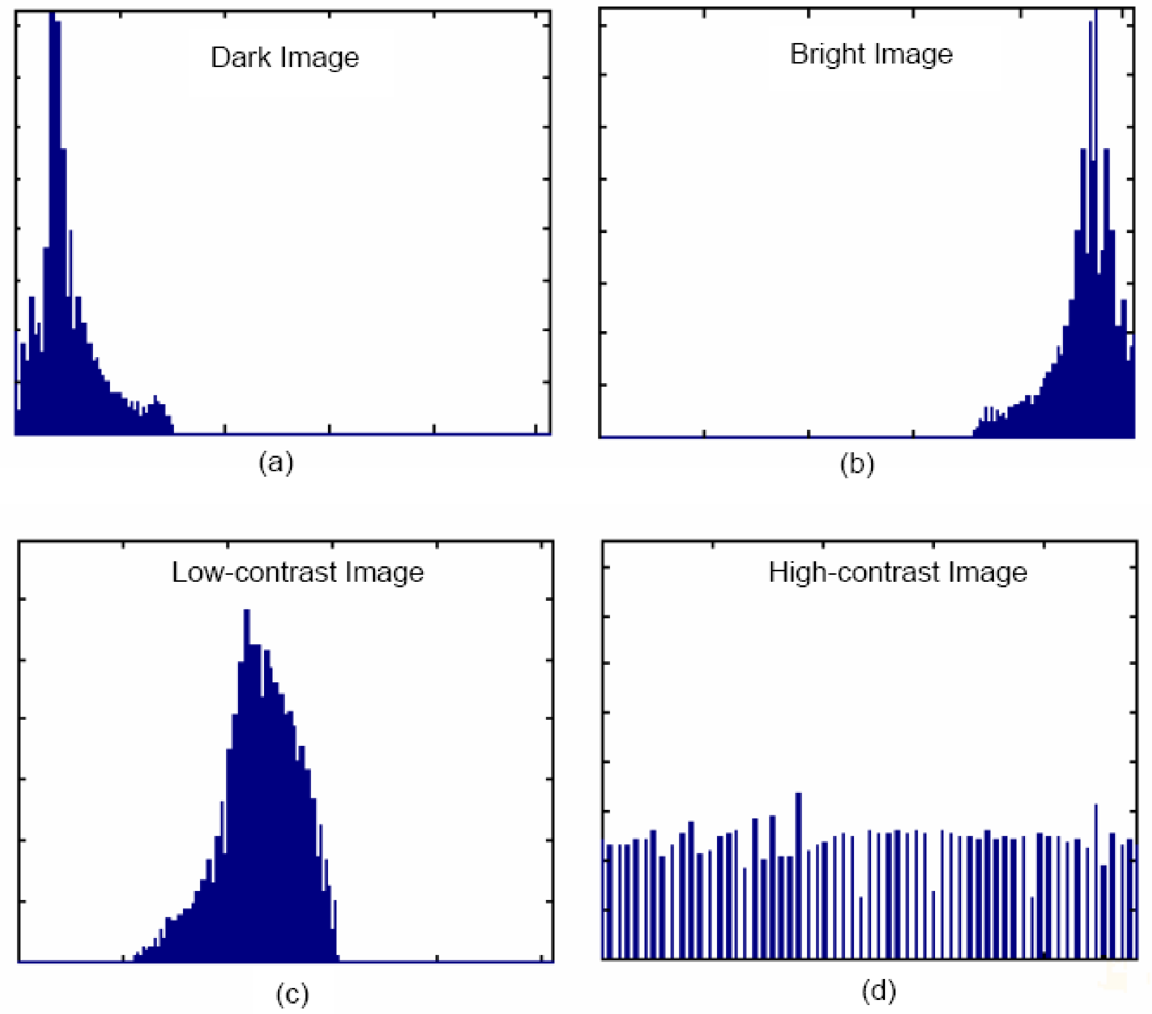


Figure 26. Histogram of Four Basic Image Types. (From: Teo, 2003)

Histogram equalization adjusts the image pixel distribution into a uniform shape so that “optimal” contrast is obtained. Such a transform insures that the image will not be too bright or too dark, which would be the case with a narrow image pixel distribution. Adaptive histogram equalization optimizes contrast enhancement based on local image data sections, and an interpolation scheme is used to minimize boundary discontinuities (Teo, 2003).

2. Contrast Limited Adaptive Histogram Equalization

The CLAHE scheme is designed to limit contrast enhancement in homogeneous regions in the image to limit noise impacts, as homogenous regions often have large number of pixels with the same gray levels. CLAHE effectively limits the amplification of noise from the homogenous regions by allowing only a fixed number of pixels in each gray level and redistributing pixels in excess of that number through the histogram through a process known as “clipping”. To that extent, CLAHE also helps to avoid saturations resulting from bright sources in the images. However, experiments showed that applying CLAHE was really beneficial when applied to images with initial low contrast properties only, and improvements were not apparent when applied to images with initial good contrast properties, as already noted by Teo (Teo, 2003). Thus CLAHE is applied in this study to night vision images only, as they have lower contrast properties than thermal images.

3. Experimentation of CLAHE with Night Vision and Thermal Images.

The author devised a simple visual experiment to test the suitability of the CLAHE algorithm in improving image contrast and its resulting impacts on fused image characteristics. First, CLAHE was applied to each night vision and thermal image, and the processed images sent through the fusion process.

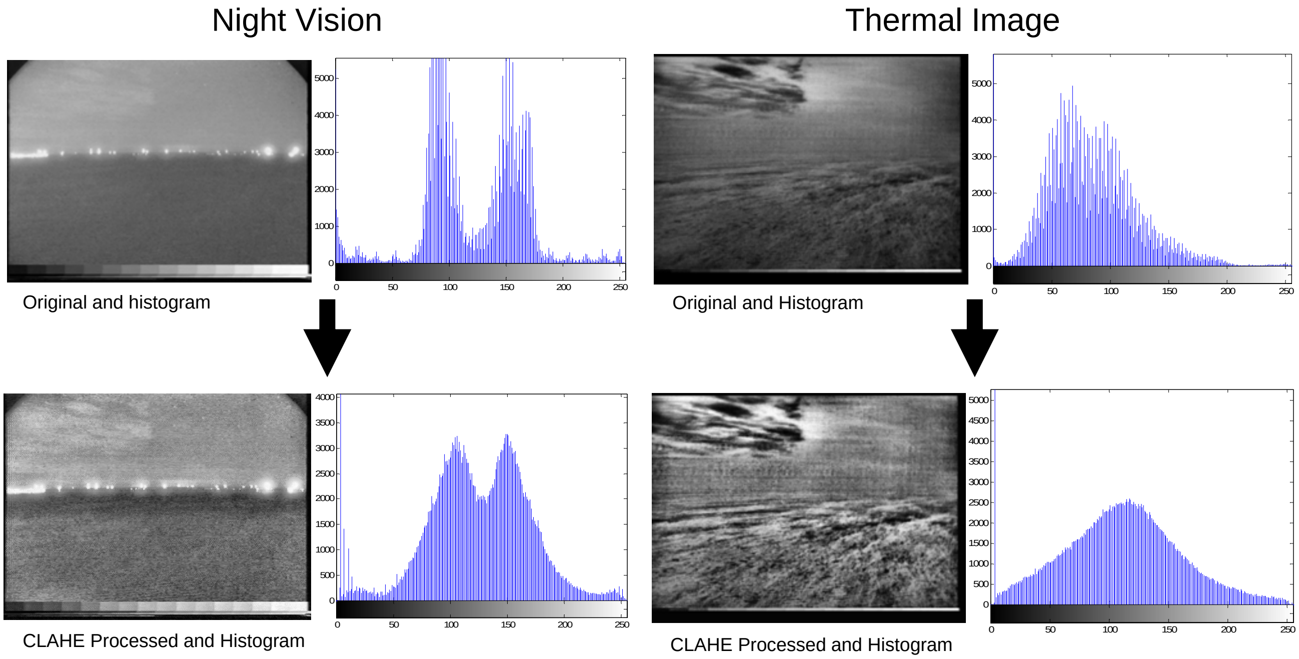
4. Observations and Comments

A few scenes were used in this experiment and CLAHE applied to both night vision and thermal images through four combinations and compared with each other. The combinations considered were

- Night vision, thermal (both non-CLAHE)
- Night vision (CLAHE), thermal (non-CLAHE)
- Night vision (non-CLAHE), thermal (CLAHE)
- Night vision, thermal (both CLAHE).

Two selected scenes’ night vision and thermal images were CLAHE processed and their histograms are shown to highlight the redistribution of the histogram in

Figure 27. Figure 28 fused image results. Results show subtle details are revealed in the processed night vision images which were not obvious in the original images. In scene A, the foreground in the night vision image exhibits a gradient of a different shade of gray as it progresses further and nearer the lights. Such a subtle difference may mean the ability to detect soldiers or a vehicle masked by the present of strong light sources. However, results also showed that there was no improvement in the thermal image as the processed image becomes cluttered with speckles in the originally gray areas.


Night Vision

Thermal Image

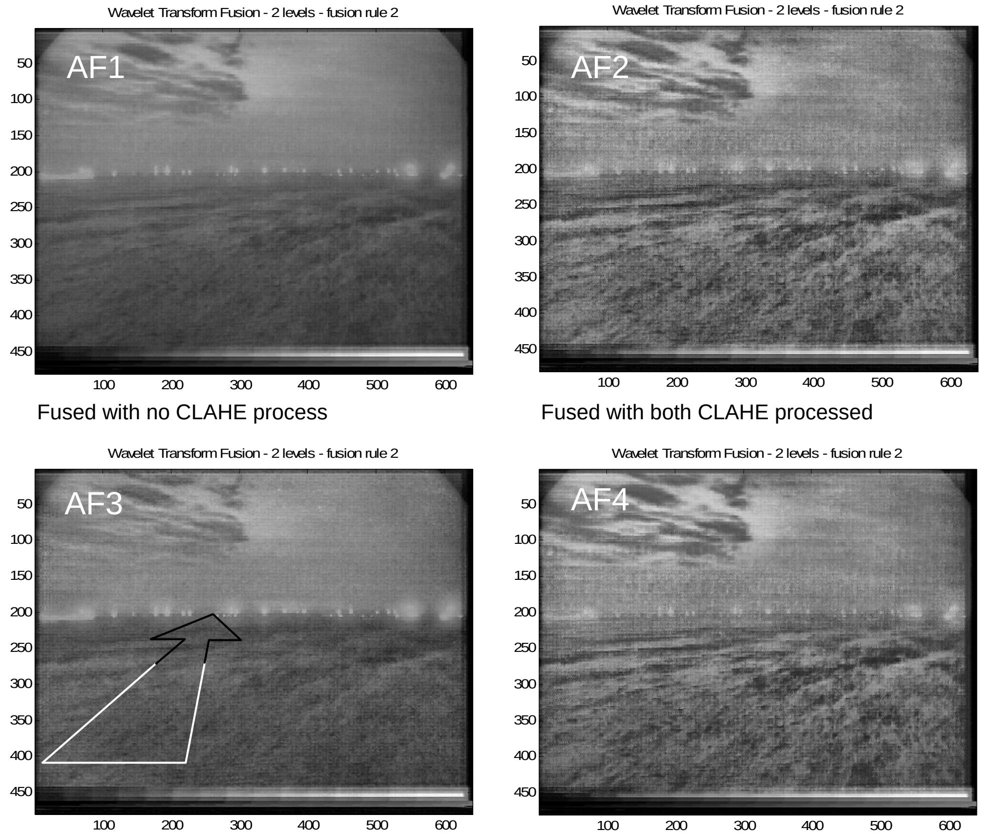
Original and histogram

Original and Histogram

CLAHE Processed and Histogram

CLAHE Processed and Histogram

Figure 27. Comparison of Original and CLAHE Processed for Scene A


Wavelet Transform Fusion - 2 levels - fusion rule 2

Wavelet Transform Fusion - 2 levels - fusion rule 2

Fused with no CLAHE process

Fused with both CLAHE processed

Wavelet Transform Fusion - 2 levels - fusion rule 2

Wavelet Transform Fusion - 2 levels - fusion rule 2

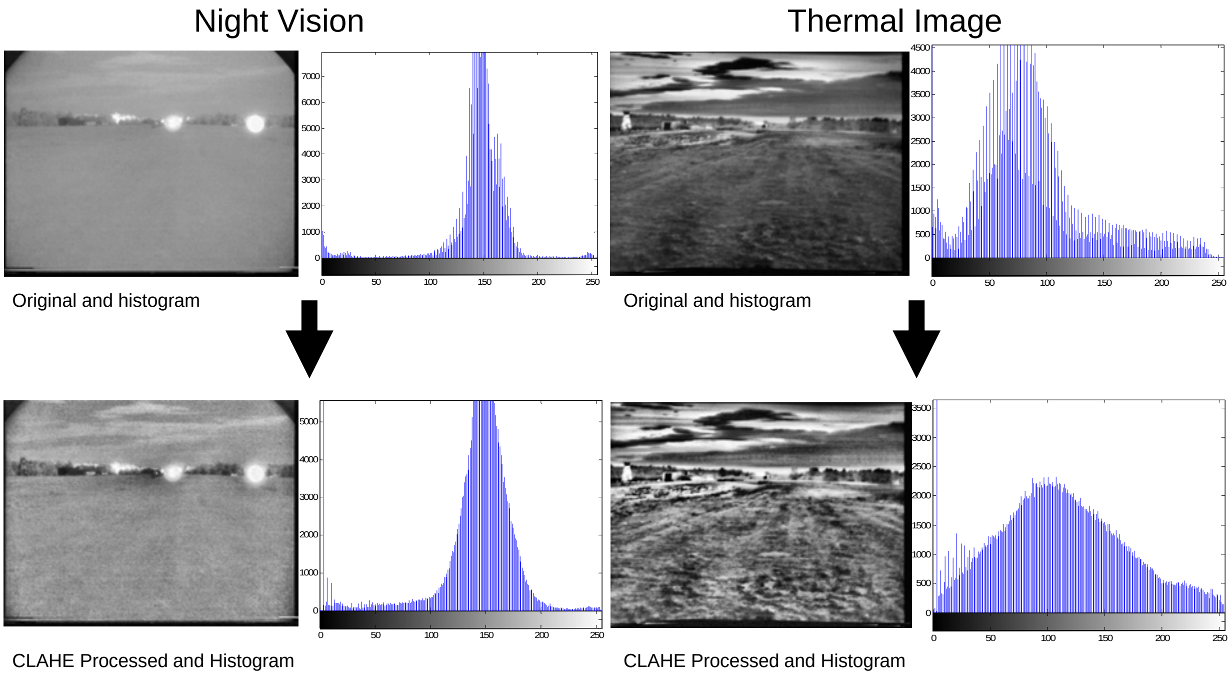
Fused with Night Vision CLAHE processed

Fused with Thermal Image CLAHE processed

Figure 28. Comparison of Series of Fused Images for Scene A

Figure 28 shows that image AF3 has a close resemblance to image AF1. Image AF3 has a subtle addition of a depth perception with the help of gray level differences as the foreground moves further towards the lights, as highlighted by the dual tone arrow shown on the figure. In addition, image AF3 has better contrast than image AF1 without significant clutter. Images AF2 and AF 4 both look cluttered as a result of enhancing the contrast

Figure 29 and 30 show the results of applying the CLAHE process to a different scene (called scene B) and the resulting fused images. Here one notices that the night vision original image has large gray areas and exhibits homogenous gray tone throughout. After CLAHE processing, clouds and foreground regions present in the night vision image exhibit subtle differences that help distinguish them. The thermal image has a distinct contrast between regions of temperature differences, which are enhanced after CLAHE processing, resulting in a processed image which looks rather “busy”.



Original and histogram

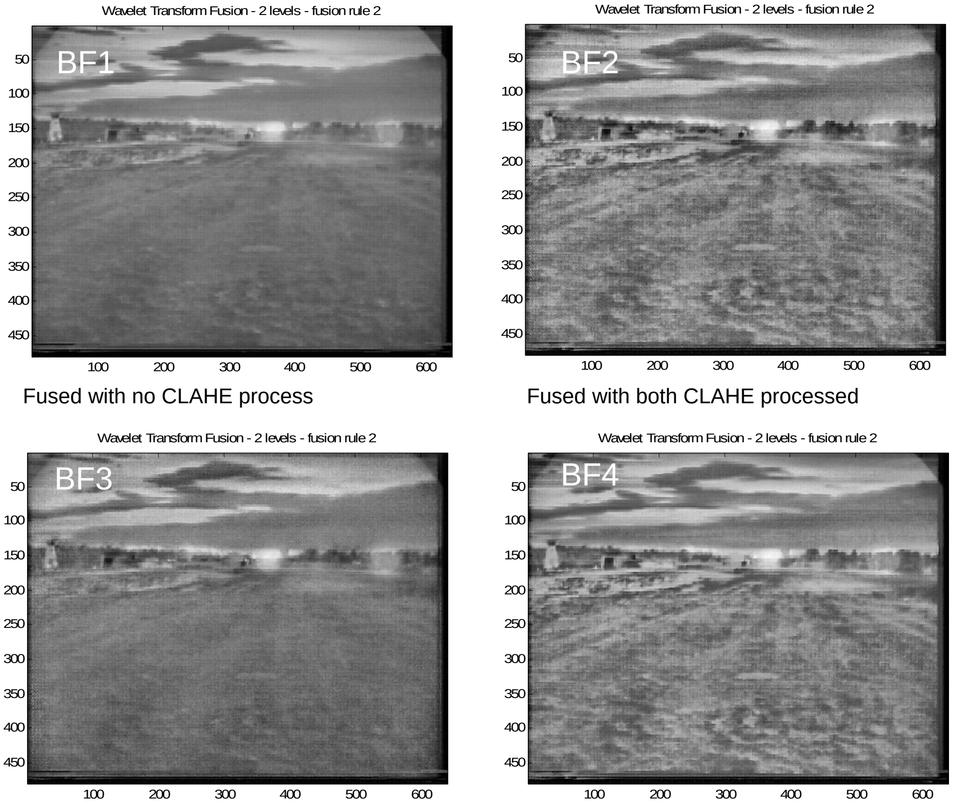
Original and histogram

CLAHE Processed and Histogram

CLAHE Processed and Histogram

Figure 29. Comparison of Original and CLAHE Processed for Scene B

In Figure 29, image BF3 displays similar results as image BF1. Images BF2 and BF4 display similar characteristics as those noted in scene A. Without looking at the original images, one may not be able to identify that the foreground of the fused image is solid ground texture but rather may potentially perceive it as some water feature. Similar comments may also be applicable to scene A. Finally, image BF3 shows the light sources relatively better than image BF1 without over-saturation.



Fused with no CLAHE process

Fused with both CLAHE processed

Fused with Night Vision CLAHE processed

Fused with Thermal Image CLAHE processed

Figure 30. Comparison of Series of Fused Images for Scene B

D. FEATURES AND OBJECTS

For a night vision image, light sources are important features that need to be retained in the fused image. This applies to heat sources in thermal images. If these features are not highlighted in the fused image, the user may not be able to differentiate where the additional features originate from. In addition, the user may not be able to analyze the fused image as effectively without such a distinction made available.

Thus, we generated overlays to be laid over the fused image which identify specific regions as originating from light sources or from heat sources, and included that capability in the GUI fusion application. Additional overlays may also be created to highlight important information or features that have been extracted or identified from the image processing prior to the fusion process. Overlays can also be used as an effective tool to annotate the fused image with information such as local image quality indices, etc... To construct the overlays that are to be used as part of the fused image, the coefficients obtained from the wavelet decomposition have to be further processed, and go through manipulations described below before being overlaid as part of the fused image.

1. Intensity Identification

Regions of interest may be spotted by scanning pixels based on a predetermined threshold value, as objects of interest in night vision and thermal are expected to exhibit high pixel value intensities. In this study, the threshold value was selected after trials and errors on several images to be equal to 200.

2. Edge Enhancement

By enhancing the edges of regions of high intensity, the author is able to separate the regions of interest from the rest of the image, which is then masked with zeros.

3. Color Depiction

To distinguish the overlays, a different color was used for each overlay. Yellow color was selected to highlight the light sources and red color selected for the heat sources. Figure 31 illustrates the process described above.

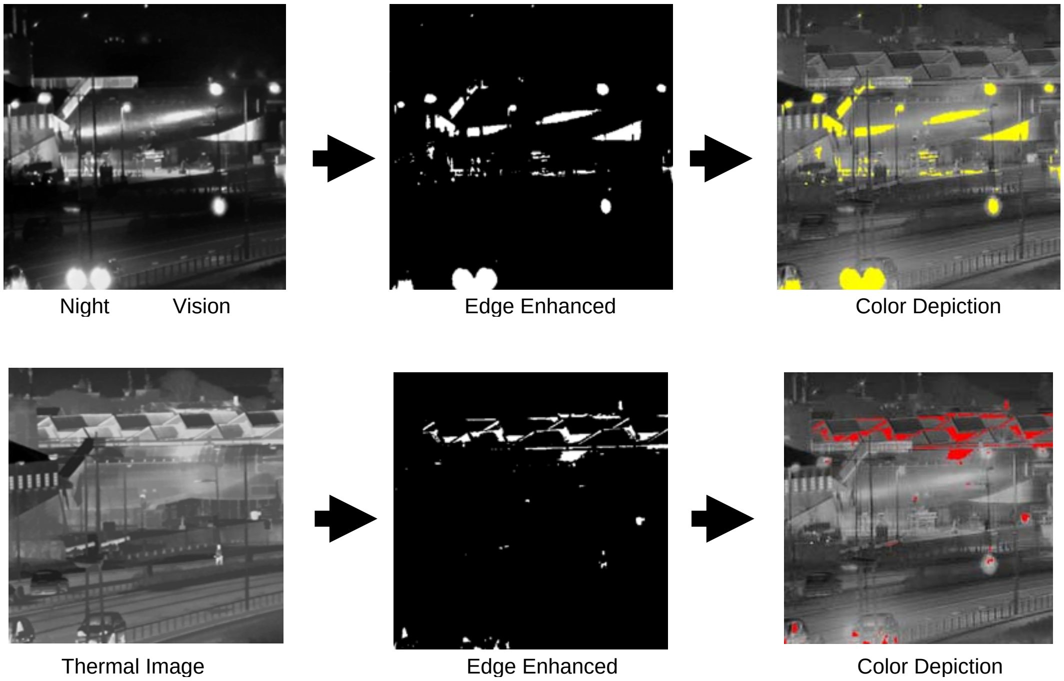

Night Vision
Edge Enhanced
Color Depiction
Thermal Image
Edge Enhanced
Color Depiction

Figure 31. Overlays Construction Process

E. IMAGE QUALITY EVALUATION

Image quality evaluation is often used in the industry to assess the performance of imaging peripherals such as digital cameras, scanners, printers and displays. It is also used extensively by software engineers who need to investigate the performance of digital image and video compression algorithms. Image quality assessment can be done in two ways, subjective and objective image assessment. Subjective image assessment relies on assessment inputs from human observers, based mainly on the overall image appearance to the observers. However, as mentioned by Wolin et al. (1998) it is difficult to distill objective data about specific image elements and attributes from image evaluation processes and procedures that rely on human observations. Quantification of image quality attributes is confounded by the inherent subjectivity of human judgment, and the fact that human perception is a complex mixture of psychology, physiology, and environment. At the same time subjective assessment is difficult to implement as it is time-consuming and requires trained subjects. Wolin et al. (1998) stated that quantification (of images) provides the basis for inter-system comparisons, evaluation of performance against specifications, and can be a critical component in process control and failure analysis. Objective image quality evaluation systems can provide the repeatability and reliability lacking in subjective processes. A machine-vision-based system can provide detailed information about individual attributes that contribute to the overall perception of image quality.

Currently there has not been an established method of quality assessment and most proposed quality assessment metrics are used mainly for color images application and for more specific application requirements. A commonly used quality assessment method is the comparison with an undistorted image. Often the undistorted image is the original image and the image for assessment is original with a lower quality. Specifically for night vision and thermal image fusion applications, the challenge is the low contrast and lack of color differentiation of the source images, as compared to fusion of colored digital images.

The method considered in our study is based on the universal quality index introduced by Wang and Bovik (2002). This universal index models any distortion as a combination of three different factors: loss of correlation, luminance distortion and contrast distortion (Wang, 2002). The method uses the original image as a comparison for the three factors. The index is a result of the product of the three factors measured between x as the original image and y as the test image defined as

$$Q = \frac{4\sigma_{xy}\overline{x}\,\overline{y}}{({\sigma_x}^2 + {\sigma_y}^2)\left[(\overline{x})^2 + (\overline{y})^2\right]}, \qquad (5.1)$$

where

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i,\ \ \overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i,\ \ \sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2,\ \ \sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \overline{y})^2,$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y}).$$

Rewriting

$$Q = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \cdot \frac{2\bar{x}\,\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \frac{2\sigma_x \sigma_y}{(\sigma_x{}^2 + \sigma_y{}^2)}, \qquad (5.2)$$

with the first component measuring the correlation coefficient of the pixel between x and y, the second component measuring the mean luminance between the two image corresponding pixels and finally the third component measuring the similarity of the contrast of x and y.

To apply this concept to an image, the approach uses a sliding window of size BxB and moves pixel-by-pixel horizontally and vertically through out the entire image from the top-left corner. At every step of the horizontal move, the local quality index within the sliding window is the average of the quality indexes inside the window.

For the fusion application, the quality index is retrieved via comparison of the fused image with both the night vision and thermal image. This method ensures that the fused image contains substantial contributions from both images. The fused image of a particular fusion scheme is deemed better over the other fusion schemes if both quality indexes are the best. The algorithm implemented in the fusion application is a Matlab code provided by Wang (2002).

THIS PAGE INTENTIONALLY LEFT BLANK

V. IMAGE FUSION APPLICATION

We introduced several wavelet families in Chapter III, the CLAHE image enhancement method in Chapter IV, and various fusion methods in Chapter III. Experiments on various images showed that not a single fusion scheme was best for all night vision and thermal images. Therefore, we investigated an iterative scheme which iteratively selected wavelet families and fusion schemes to derive the 'best' fused image for a specific image pair. This chapter describes the various processes involved in that task.

A. APPLICATION PROCESS FLOW

Experiments showed that an overall specific image fusion method couldn't be identified for all images considered. Note that identifying the "ideal" or "best" fused image may be done via human evaluation which relies on human perception. However such an evaluation method is tedious, time-consuming, and results may vary from person to person. A computational evaluation would be more accurate, consistent and may be easier to implement in the fusion application.

As described in Chapter III, Daubechies, Coifman wavelet and Symlet wavelets were the wavelet families best suited for edge detection. As a result, we considered different basis functions from these wavelet families and identified which one led to he "best" fused image for each pair of images considered. The basis functions considered in our study were db2, db5, db7, db9, coif2, coif5, sym2 and sym5. For each image, we run the image fusion scheme through each wavelet basis function, using the region-based fusion scheme (fusion rule 3, described earlier in Chapter III, section A.5.c, and computed the image quality index. Figure 32 illustrates the general fusion application process flow.

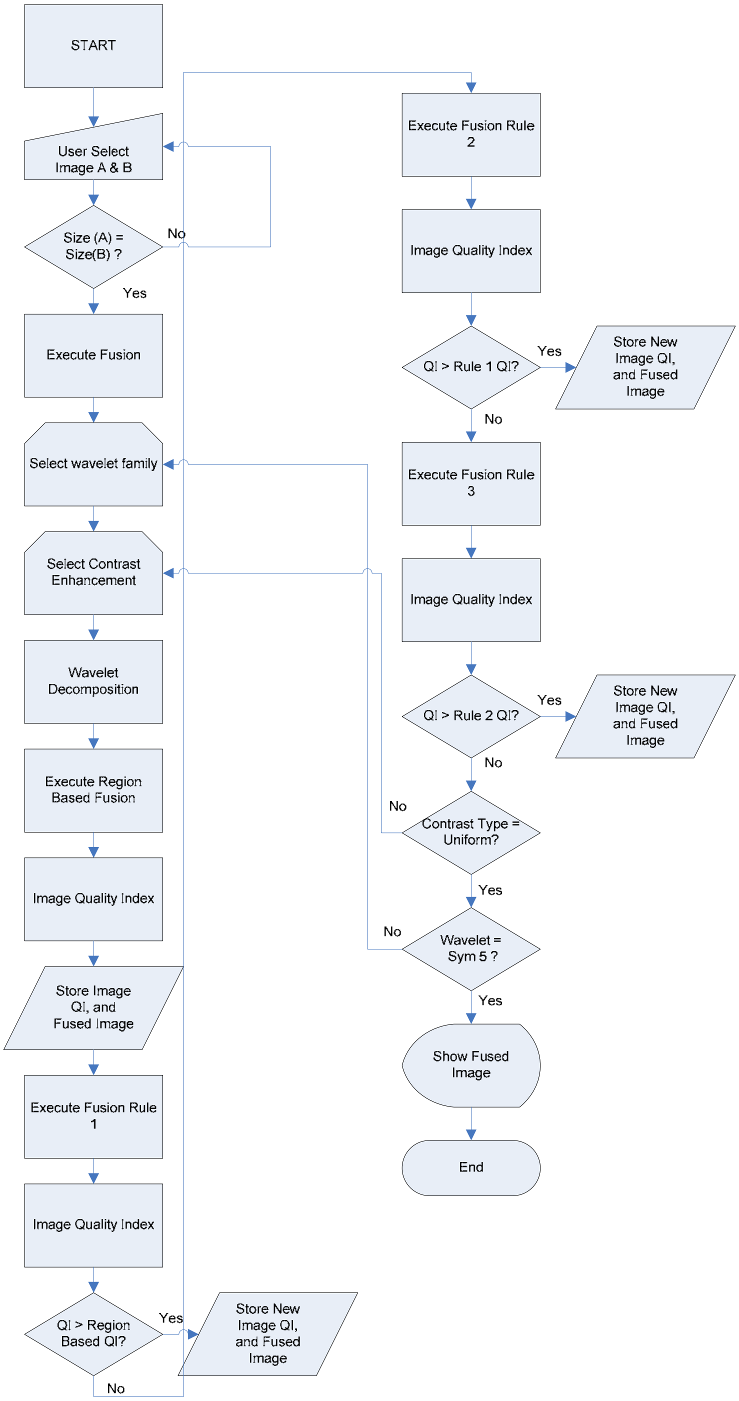


Figure 32. General Fusion Process Application Flow

In the application, some portions of the program were obtained from the earlier Chow study. Image manipulation and identification of high intensity regions, image edge detection, displaying of morphological images and adjustments of image and wavelet attributes extensively use software from the MATLAB image processing toolbox and the morphology toolbox from the SDC Morphology Toolbox (2006).

B. GUI APPLICATION

A MATLAB-based graphic user interface (GUI) application was designed and implemented to execute fusion of images efficiently and present the output from the fusion process to the user. The GUI application makes the process of experimenting with the fusion of the images easier and faster. The GUI application goes through an iterative process in terms of selection of best wavelet transform fusion rule or scheme. Other features of the GUI includes selection of images types, execution of fusion, information display and used of overlays help to provide the user with information and selection of display format. The codes programmed for this GUI are contained in Appendix A and the detail GUI application flowchart is contained in Appendix B.

1. Image Selection and Execution of Fusion

The selection of image type and image files is done through a dialogue window that previews the images under consideration. The user may carry out the fusion process by pressing the “Execute” button once the selection of the images fit to some predefined criteria. These criteria are as follows:

- Images selected for the fusion process must have one of the following formats: JPEG, TIFF, PNG and BMP.
- Both images selected must be of the same dimensions by comparing the second file selection with the first. A re-selection of the second file is needed if it does not conform to the first file.
- The Matlab wavelet toolbox at the present is only able to process grayscale images, and a conversion is applied if the image is not grayscale. However, note that most night vision and thermal

images offer dual tones of black and white or black and green so minimum alteration is done to the image attributes.

- The user may reselect all images by pressing “Clear”.
- The information about the images such as filenames and image dimensions are displayed on the GUI once the selections have been made in the “Image details” panel.
- Finally, all unrelated functions at each stage of the application from selection of images to post-processing display are “dimmed” making them not selectable to prevent unnecessary or incorrect processing.

2. Information Display

Once the fusion process is completed and the optimal fusion scheme is selected, the information related to the process such as fusion wavelet family, contrast distribution selected, quality index relating to night vision image, quality index relating to thermal image and the selected fusion scheme are displayed in the “Image Fusion Information” panel. Recall that the quality index relating to the images is the value that is determined using the universal quality index used for image evaluation.

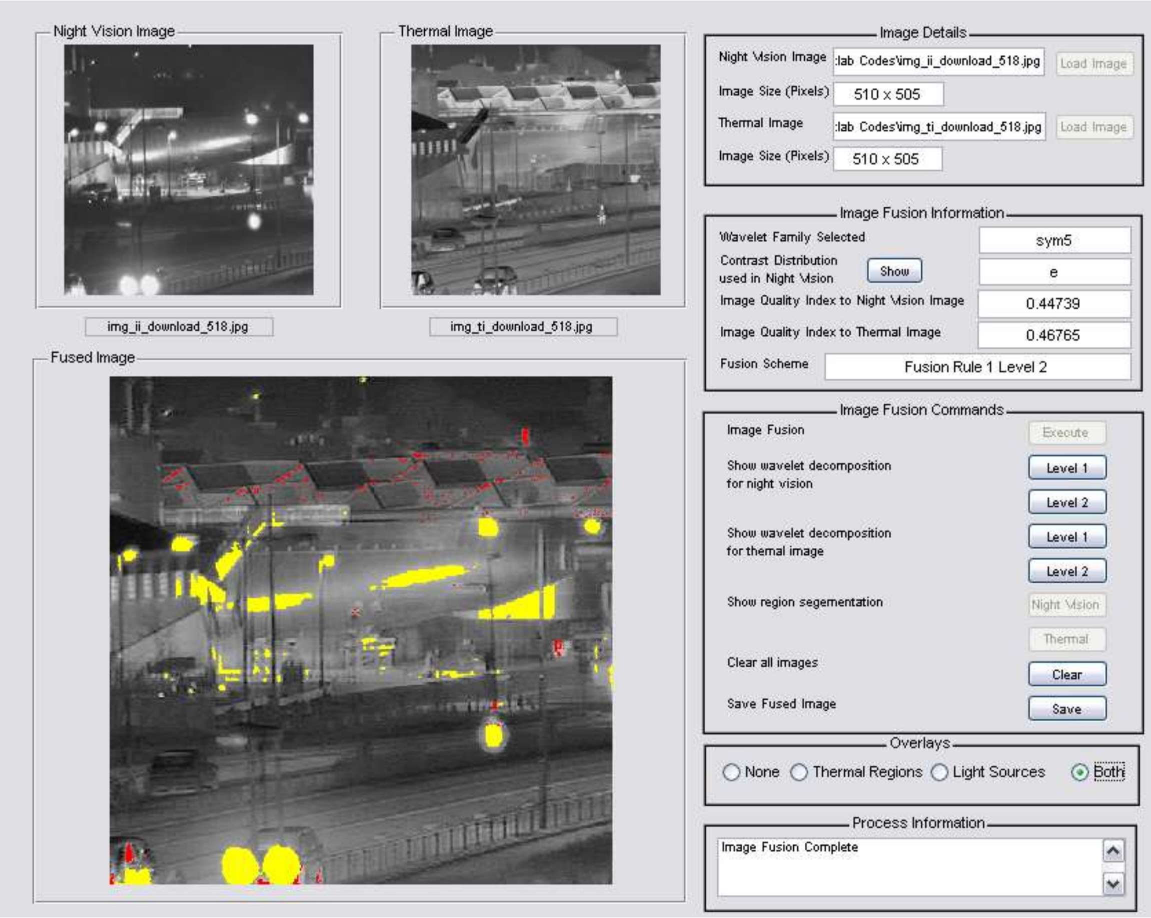
3. Wavelet Decomposition Process

Since this fusion application uses wavelet transformation fusion, the wavelet decomposition at each level contains important information that the user may be interested in. The various levels of decomposition are available to the user for inspection when the decomposition level buttons relating to the night vision or thermal image are pressed.

The GUI is set up to allow the user to view the segmented regions for night vision and the thermal images by pressing the corresponding buttons in the “Image Fusion Commands” panel when region-based fusion is selected. Finally, the fused image with overlays may be saved, by clicking on the “Save” button.

4. Image Overlays

As part of the post-processing needed for image fusion, the overlays help to depict critical information extracted from the night vision and thermal images, as mentioned in the previous section. The mode of display of the fused image can be changed by the user. Figure 33 shows a screen shot of a fused image with both overlays. Note that the default display of the fused image when the fusion process is complete is “None” which corresponds to the fused image without overlays. The user may choose overlays which represent “Light Sources” “Heat Sources” or “Both” as preferred.



img_ii_download_518.jpg

img_ti_download_518.jpg

Night Vision Image: :lab Codes\img_ii_download_518.jpg

Image Size (Pixels): 510 x 505

Thermal Image: :lab Codes\img_ti_download_518.jpg

Image Size (Pixels): 510 x 505

Wavelet Family Selected: sym5

Contrast Distribution used in Night Vision: e

Image Quality Index to Night Vision Image: 0.44739

Image Quality Index to Thermal Image: 0.46765

Fusion Scheme: Fusion Rule 1 Level 2

Image Fusion

Show wavelet decomposition for night vision

Show wavelet decomposition for thermal image

Show region segementation

Clear all images

Save Fused Image

○ None ○ Thermal Regions ○ Light Sources ● Both

Image Fusion Complete

Figure 33. Screenshot of GUI Application with Fused Image and Overlays

C. UNDERLYING PROCESSES

As previously mentioned in Chapter III, the watershed transform is used to segment the wavelet-decomposed image into regions. As proposed, the ideal number of regions is about forty. As this segmentation is controlled by the threshold setting for the marker-based watershed transform, one needs to vary the threshold settling to achieve the correct number of region segmentations. Increasing the threshold leads to fewer regions and vice versa. Note that there is no straight forward way of obtaining the ideal number of segmentation regions as region segmentations are dependent on the composition of the image. Therefore, a loop implemented to adjust the threshold according to the number of regions segmented until the region segmentations fell below forty for the region-based fusion scheme. Figure 34 illustrates the threshold adjustment and the loop process flowchart.

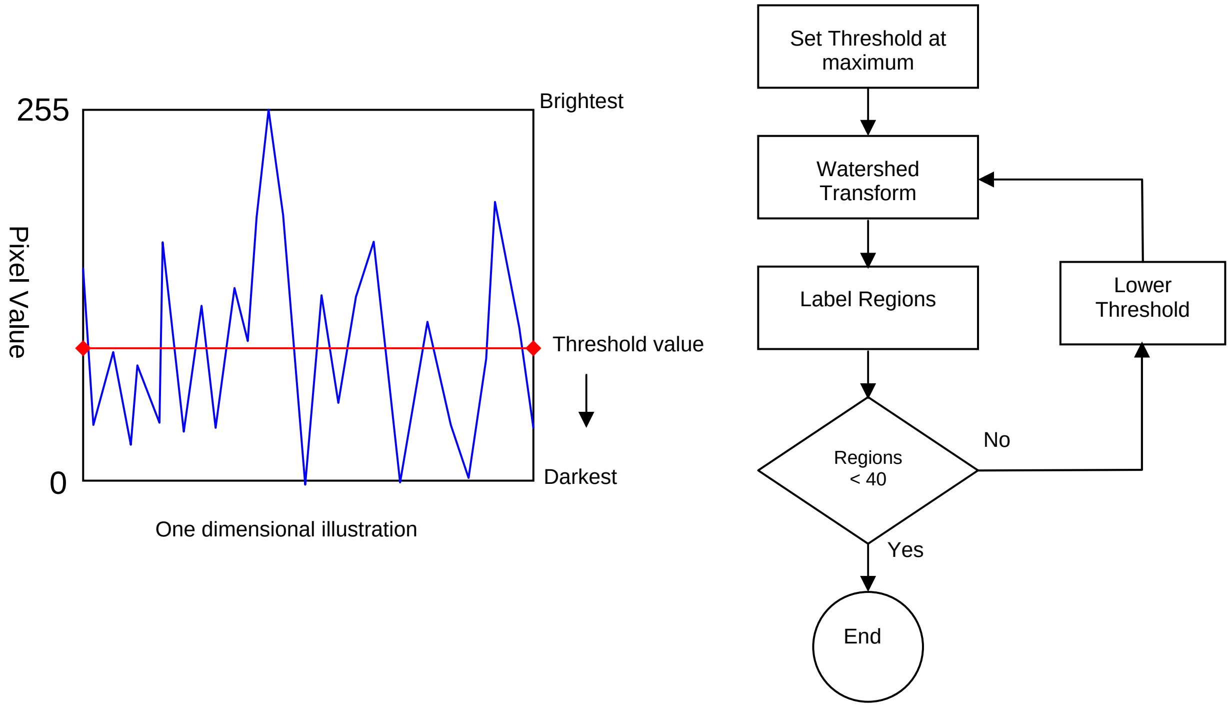
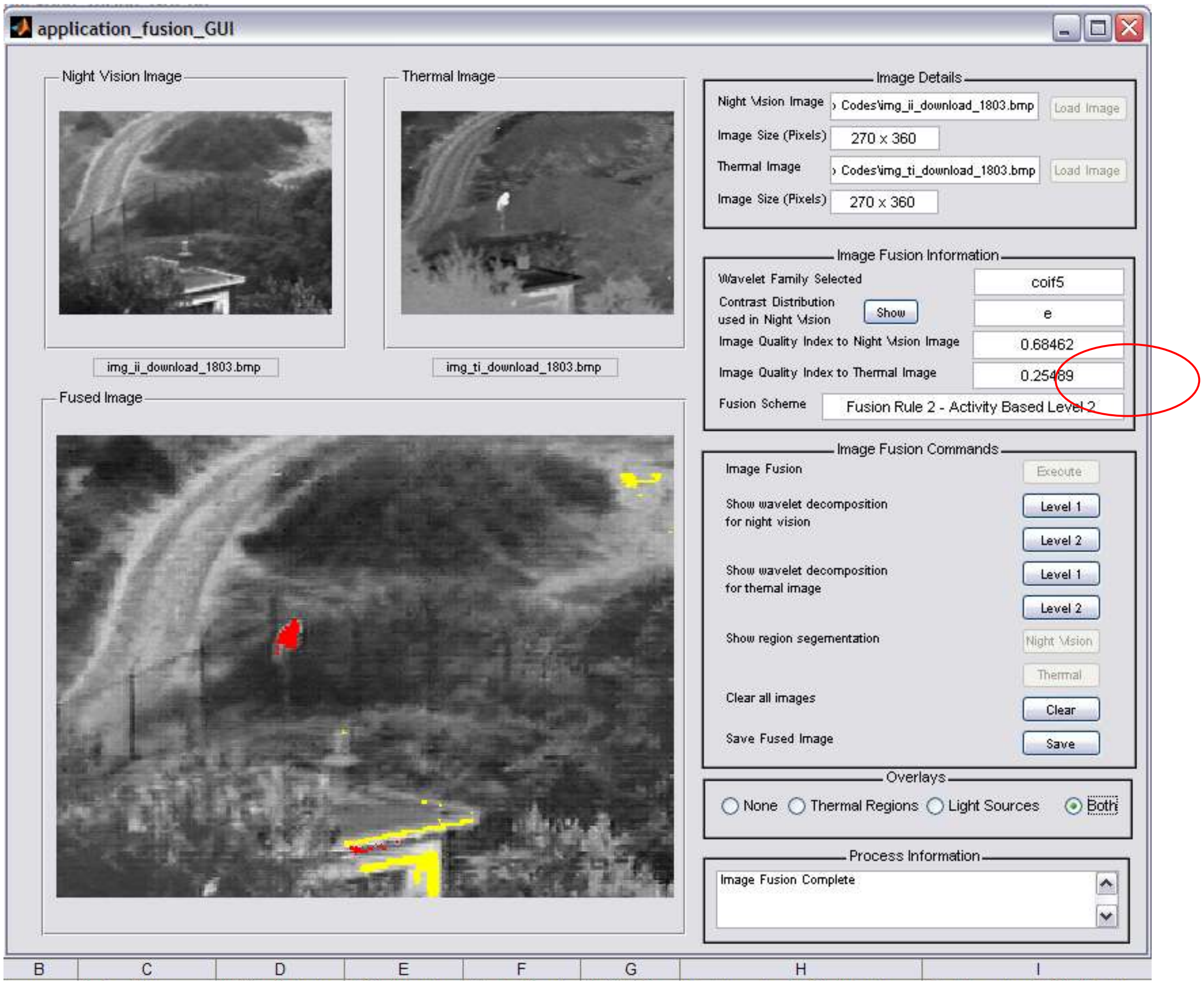


Figure 34. a) Threshold Adjustment, b) Loop Process Flowchart

D. RESULTS AND DISCUSSION

The GUI application was used to evaluate three images available from the Naval Research Laboratory. The fused images and its image quality indexes were examined to determine whether the image with the best image quality index was selected. From the 160 permutations that were collected for each image, the application was able to select the image with a quality index that is among the top ten percentages of the night vision quality index. However, results obtained for the thermal image quality index were mixed as the corresponding ranking fluctuated between 33 and 76. Additional work is needed on a larger set of images and the fusion selection process needs to be optimized. The complete data set for the experiment on the image shown in Figure 35 is available in Appendix C.



img_ii_download_1803.bmp

img_ti_download_1803.bmp

Image Details

Night Vision Image: › Codes\img_ii_download_1803.bmp

Image Size (Pixels): 270 x 360

Thermal Image: › Codes\img_ti_download_1803.bmp

Image Size (Pixels): 270 x 360

Image Fusion Information

Wavelet Family Selected: coif5

Contrast Distribution used in Night Vision: e

Image Quality Index to Night Vision Image: 0.68462

Image Quality Index to Thermal Image: 0.25489

Fusion Scheme: Fusion Rule 2 - Activity Based Level 2

Image Fusion

Show wavelet decomposition for night vision

Show wavelet decomposition for thermal image

Show region segementation

Clear all images

Save Fused Image

Overlays

○ None ○ Thermal Regions ○ Light Sources ● Both

Process Information

Image Fusion Complete

B	C	D	E	F	G	H	I
Image	Fusion Scheme	NV Quality Index	TI Quality Index	Wavelet Family	CLAHE Type	Rank based on TI Quality Index	Rank based on TI Quality Index
1803	Fusion Rule 2	0.68666	0.24729	'sym5'	'e'	1	91
1803	Fusion Rule 2	0.68462	0.25489	'coif5'	'e'	2	76
1803	Fusion Rule 2	0.68148	0.25083	'db5'	'e'	3	84
1803	Fusion Rule 2	0.6812	0.25432	'coif2'	'e'	4	78
1803	Fusion Rule 2	0.68009	0.31119	'sym5'	'r'	5	48

Figure 35. Experiment Test Results for GUI Application.

VI. CONCLUSION AND RECOMMENDATIONS

A. SUMMARY

This study investigated wavelet transform fusion applied to night vision and thermal images and three different fusion schemes were considered. A MATLAB-based GUI tool was designed which combined the various schemes considered in this study and allowed the user to investigate image fusion conveniently and efficiently.

Region-based fusion exploits the features and objects within a scene. By identifying regions of interest, the author is able to incorporate these unique features from each type of image to be merged in the final image.

The fusion scheme developed especially for region-based fusion retains most of the features of both images. However, results seen on the limited set of images considered in this study seem to indicate that the fusion scheme may not necessarily produce the “best” fused image, and further research is needed to continue the investigations. In addition, investigations showed that the image composition plays a crucial role in the ability to extract information. By using digital image processing and the morphological method, images can be enhanced to bring out features and information can be extracted to highlight its importance or to allow flexibility in the resulting display.

B. RECOMMENDATION FOR FURTHER RESEARCH

1. Counterchecking of Fusion Application with Human Visual Evaluation

Our initial results need to be validated by conducting a human subject test study to verify that the image fusion application has the ability to select the best fusion scheme based on the selected image quality evaluation. Furthermore, additional images should be collected and used in such a testing phase to achieve an accurate evaluation of the fusion application performances

2. Other Image Quality Evaluation Methods

The selected image quality evaluation index is based on a comparison of the original images to the fused image. This approach may be a computational challenge when such implementation is done on hardware. Thus, a more computationally efficient non-comparison method of image evaluation should be explored in follow-on work.

3. Real- Time Implementation

The fusion application uses registered images for experimentation. In an attempt to mimic hardware implementation, the fusion application may be developed to retrieve inputs from night vision devices and thermal images and thereafter perform a real time fusion. At that point, the main challenge lies in the registration of the inputs of the night vision and thermal imagers as well as the computation power of the platform which the fusion application processed, and such issues would need to be considered in follow-on work.

APPENDIX A. MATLAB ALGORITHMS

This Appendix contains the following MATLAB sources files:

1. Wavelet Transformation Fusion Application

   (mat_application_wavelet_trans_fusion.m)

2. Wavelet Transformation Fusion Application with GUI incorporated

   application_fusion_GUI.m

1. WAVELET TRANSFORMATION FUSION APPLICATION

```
% M file : mat_application_wavelet_trans_fusion.m
% This program executes the fusion of Night Vision and Thermal Images using
% wavelet transformation. This program has not been incorporated with GUI
% and serve as a basic function before incorporating with the GUI.
% The program uses an iterative method to go through a list of wavelet family
% to find the ideal family for the fusion. At the same time, the images are
% enhanced prior to executing the wavelet transform this helps to increase
% the separation of the regions of high and low intensity. Further
% processing on the wavelet coefficents is done, where the data is used to
% segment the image into regions of interest. These regions are used at a
% later stage for better illustration to the user when the fused image is
% presented. The quality of fused image is quantified using a quality index
% compared with the original images. This quality index determines the
% selection of the ideal wavelet family and the amount of enhancement to be
% carried out.

% Application start
close all;
clear all;

%---Open images------------------------------------------------------------
% Open Night Vision Image
[img_nv,path_nv] = uigetfile({'*.bmp';'*.jpg';'*.hdf';'*.png';'*.tif';'*.*'},'Select Night Vision Image');

if isequal(img_nv,0)                          % check image selection, if there is no selection
display msg
   disp('No image selected')
else
   disp(['Image Selected ', fullfile(path_nv, img_nv)])         % check image selection, if selected,
check file image extension
[nv_pathstr, nv_name, nv_ext, nv_versn] = fileparts(img_nv)      % supported formats .bmp, .jpg,
.hdf, .png, .tif
    switch lower(nv_ext)
     case '.bmp'
         img_nv_ext = nv_ext;                          % store file extension
     case '.jpg'
```

```
         img_nv_ext = nv_ext;
     case '.hdf'
          img_nv_ext = nv_ext;
     case '.png'
          img_nv_ext = nv_ext;
     case '.tif'
          img_nv_ext = nv_ext;
     otherwise
       disp('Image type is not supported')
    end
    % Convert to grayscale if needed
    [A,mapA]=imread(fullfile(path_nv, img_nv));                    % read image and stores in matrix A
    [a1,a2,a3]= size(A);                                       % If image is rgb format convert to grayscale
    if isequal(a3,3)
    G1 = rgb2gray(A);
    [A,mapA] = gray2ind(G1,256);
    end
end

% Open Thermal Image
[img_ti,path_ti] = uigetfile({'*.bmp';'*.jpg';'*.hdf';'*.png';'*.tif';'*.*'},'Select Thermal Image');

if isequal(img_ti,0)                                   % check image selection, if there is no selection
display msg
   disp('No image selected')
else
   disp(['Image Selected ', fullfile(path_ti,img_ti)])        % check image selection, if selected, check
file image extension
[ti_pathstr, ti_name, ti_ext, ti_versn] = fileparts(img_ti)     % supported formats .bmp, .jpg, .hdf,
.png, .tif
    switch lower(ti_ext)
     case '.bmp'
         img_ti_ext = ti_ext;                            % store file extension
     case '.jpg'
         img_ti_ext = ti_ext;
     case '.hdf'
          img_ti_ext = ti_ext;
     case '.png'
          img_ti_ext = ti_ext;
     case '.tif'
          img_ti_ext = ti_ext;
     otherwise
       disp('Image type is not supported')
    end
end

% Convert to grayscale if needed
[B,mapB]=imread(fullfile(path_ti,img_ti));                    % read image and stores in matrix A
[b1,b2,b3]= size(B);                                     % If image is rgb format convert to grayscale
if isequal(b3,3)
    G2 = rgb2gray(B);
    [B,mapB] = gray2ind(G2,256);
end

% Check if image 1 and image 2 dimensions are same size
```

```
if isequal(size(A),size(B))&& isequal(img_nv_ext,img_ti_ext)
   disp('Click Execute for Fusion Processing')
else
   disp('Image selected do not have the same dimensions as Night Vision Image or same format
please reselect TI image')
   error_msg = 1;
end

%---End file opening portion---------------------------------------------

% Initialise quality evaluation parameters
dbase_II_qi = -1;
dbase_TI_qi = -1;

%---Start Fusion----------------------------------------------------------
% Parameters for image enhancement, define tile sizes and variables
N = 2;                                        % tile size set at smaller value has better clarity
M = 2;                                        % tile size set at smaller value has better clarity
data1 = ['uniform    ';'exponential';'rayleigh   '];          % store types of distribution for adapthisteq
function
celldata1 = cellstr(data1);                         % converts strings to char
strings1 = char(celldata1);

% Parameters for wavelet selection
data2 = ['db2  ';'db5  ';'db7  ';'db9  ';'sym2 ';'sym5 '...
   ;'coif2';'coif5'];                            % store types of distribution for adapthisteq function
celldata2 = cellstr(data2);                         % converts strings to char

% Enhance image
for j=1:4
   if (j<=3)
      A_con_enh = adapthisteq(A,'NumTiles',[N M],'Distribution',strings1(j));
   else
      A_con_enh = A;
   end
% Wavelet Selection
   for m=1:8
   mywave = char(celldata2(m));                          % Wavelet selection

% Wavelet decomposition (2 levels) and extraction of coefficients from
% reference on wavelet transform fusion is made to fusion.m (DEMO-Wavelets
% and Image Fusion Centrum voor Wiskunde en Informatica (CWI) -
% Paul.de.Zeeuw@cwi.nl and Region Based Dec 04 - Khin Choong Chow
   [AC,AS]=wavedec2(A_con_enh,2,mywave);
   [BC,BS]=wavedec2(B,2,mywave);

% Extract approximation coefficients of A and B
   AcA2=appcoef2(AC,AS,mywave,2);
   BcA2=appcoef2(BC,BS,mywave,2);
   AcA1=appcoef2(AC,AS,mywave,1);
   BcA1=appcoef2(BC,BS,mywave,1);

% Extract detail coefficients  distinguished as horizontal ('h'), vertical
% ('v') and diagonal ('d') details.
```

```
   AcH2=detcoef2('h',AC,AS,2);
   AcV2=detcoef2('v',AC,AS,2);
   AcD2=detcoef2('d',AC,AS,2);
   AcH1=detcoef2('h',AC,AS,1);
   AcV1=detcoef2('v',AC,AS,1);
   AcD1=detcoef2('d',AC,AS,1);

   BcH2=detcoef2('h',BC,BS,2);
   BcV2=detcoef2('v',BC,BS,2);
   BcD2=detcoef2('d',BC,BS,2);
   BcH1=detcoef2('h',BC,BS,1);
   BcV1=detcoef2('v',BC,BS,1);
   BcD1=detcoef2('d',BC,BS,1);

%---Fusion of approximate image using to region fusion rule----------------
% Boundary detection. This is done using morphological gradient. Make use
% of dilation and erosion. This is done for all the three levels.
   AgA1=mmgradm(AcA1);
   AgA2=mmgradm(AcA2);

   BgA1=mmgradm(BcA1);
   BgA2=mmgradm(BcA2);

% The mmgradm images from the two sources are combined together to form a
% universal mast for subsequent fusion purposes.
   x1=mmunion(AgA1,BgA1);
   x2=mmunion(AgA2,BgA2);

% Execute marker-based watershed algorithm
% Watershed algorithm is performed on this mast to perform a marker-based
% region segmentation. Threshold level selected for image 13 for
% segmentation. A loop is used to limit the number of regions to
% approximately to 40.
   mh1=MMREGMIN(x1,mmsecross,'dynamics');
   mh2=MMREGMIN(x2,mmsecross,'dynamics');
   nblobs1 = 0;
   nblobs2 = 0;
   level1 = 200;
   level2 = 200;
   while (nblobs1<=38)
      level1 = level1-1;
      ws1=mmcwatershed(x1, mmthreshad(mh1, uint8(level1)), mmsecross, 'regions');
      ws1c=mmcwatershed(x1, mmthreshad(mh1, uint8(level1)));
      nblobs1=mmstats(ws1,'max');                        % To determine the maximum number of
regions for each subsequent level of decomposition
   end
   while (nblobs2<=36)
      level2=level2-1;
      ws2=mmcwatershed(x2, mmthreshad(mh2, uint8(level2)), mmsecross, 'regions');
      ws2c=mmcwatershed(x2, mmthreshad(mh2, uint8(level2)));
      nblobs2=mmstats(ws2,'max');                        % To determine the maximum number of
regions for each subsequent level of decomposition
   end
% Regions are labelled
   L1=mmlabel(ws1c); L2=mmlabel(ws2c);
```

```
% To determine the area of each region
  area1=MMBLOB(ws1,'area');
  area2=MMBLOB(ws2,'area');

% Make new approximation coefficients using regionfusion.m. DD1 represents
% the decision maps.
  [Fused_cA1,limit1,DD1] = regionfusion(AcA1,BcA1,ws1,area1,nblobs1);

  [Fused_cA2,limit2,DD2] = regionfusion(AcA2,BcA2,ws2,area2,nblobs2);

% Make new detail coefficients using atb.m. Dh2 represents the decision maps.
[cH2, Dh2]=atb(AcH2, BcH2);
[cV2, Dv2]=atb(AcV2, BcV2);
[cD2, Dd2]=atb(AcD2, BcD2);
[cH1, Dh1]=atb(AcH1, BcH1);
[cV1, Dv1]=atb(AcV1, BcV1);
[cD1, Dd1]=atb(AcD1, BcD1);

% The processed approximation and detail coefficients are assembled into C.
C = fillC( Fused_cA2, cH2, cV2, cD2, cH1, cV1, cD1, AS);

% 2-levels reconstruction for the fusion of A and B.
F=waverec2(C,AS,mywave);
fusion_scheme = 'Region Based Fusion';

% Obtain Quality Index based on Night Vision Image
[II_qi II_qi_map] = img_qi(double(A), F);
% Obtain Quality Index based on Thermal Image
[TI_qi TI_qi_map] = img_qi(double(B), F);

  if(II_qi>dbase_II_qi)&&(TI_qi>dbase_TI_qi)
    dbase_II_qi = II_qi;
    dbase_TI_qi = TI_qi;
    dbase_F_img = F;
    dbase_mywave = mywave;
    if (j<=3)
      dbase_contrast = strings1(j);
    else
      dbase_contrast = 0;
    end
    dbase_fusion_scheme = fusion_scheme;
  else
  end                                 % end for if statement

%---Fusion Rule 1 Weighted Average - Level 2-------------------------------
% Average approximation coefficients.
 cA2=ave(AcA2, BcA2);
 cA1=ave(AcA1, BcA1);

% Make new detail coefficients using atb.m. Dh2 represents the decision maps.
 [cH2, Dh2]=atb(AcH2, BcH2);
 [cV2, Dv2]=atb(AcV2, BcV2);
 [cD2, Dd2]=atb(AcD2, BcD2);
```

```
  [cH1, Dh1]=atb(AcH1, BcH1);
  [cV1, Dv1]=atb(AcV1, BcV1);
  [cD1, Dd1]=atb(AcD1, BcD1);

% The processed approximation and detail coefficients are assembled into C.
  C = fillC( cA2, cH2, cV2, cD2, cH1, cV1, cD1, AS);

% 2-levels reconstruction for the fusion of A and B.
  F=waverec2(C,AS,mywave);
  fusion_scheme = 'Fusion Rule 1 Level 2';
% Obtain Quality Index based on Night Vision Image
  [II_qi II_qi_map] = img_qi(double(A), F);

% Obtain Quality Index based on Thermal Image
  [TI_qi TI_qi_map] = img_qi(double(B), F);

   if(II_qi>dbase_II_qi)&&(TI_qi>dbase_TI_qi)
      dbase_II_qi = II_qi;
      dbase_TI_qi = TI_qi;
      dbase_F_img = F;
      dbase_mywave = mywave;
      if (j<=3)
         dbase_contrast = strings1(j);
      else
         dbase_contrast = 0;
      end
      dbase_fusion_scheme = fusion_scheme;
   else
   end                                               % end for if statement
%---Fusion Rule 2 Activity Based- Level 2------------------------------------
% Average approximation coefficients.
  cA2=ave(AcA2, BcA2);
  cA1=ave(AcA1, BcA1);

% Make new detail coefficients using atb.m. Dh2 represents the decision maps.
  [cH2, Dh2]=atb_activity(AcH2, BcH2);
  [cV2, Dv2]=atb_activity(AcV2, BcV2);
  [cD2, Dd2]=atb_activity(AcD2, BcD2);
  [cH1, Dh1]=atb_activity(AcH1, BcH1);
  [cV1, Dv1]=atb_activity(AcV1, BcV1);
  [cD1, Dd1]=atb_activity(AcD1, BcD1);

% The processed approximation and detail coefficients are assembled into C.
  C = fillC( cA2, cH2, cV2, cD2, cH1, cV1, cD1, AS);

% 2-levels reconstruction for the fusion of A and B.
  F=waverec2(C,AS,mywave);
  fusion_scheme = 'Fusion Rule 2 - Activity Based Level 2';

% Obtain Quality Index based on Night Vision Image
  [II_qi II_qi_map] = img_qi(double(A), F);

% Obtain Quality Index based on Thermal Image
  [TI_qi TI_qi_map] = img_qi(double(B), F);
```

```
    if(II_qi>dbase_II_qi)&&(TI_qi>dbase_TI_qi)
        dbase_II_qi = II_qi;
        dbase_TI_qi = TI_qi;
        dbase_F_img = F;
        dbase_mywave = mywave;
        if (j<=3)
            dbase_contrast = strings1(j);
        else
            dbase_contrast = 0;
        end
        dbase_fusion_scheme = fusion_scheme;
    else
    end
%---Fusion Rule 2 Window Based- Level 2------------------------------------
% Average approximation coefficients.
  cA2=ave(AcA2, BcA2);
  cA1=ave(AcA1, BcA1);

% Make new detail coefficients using atb.m. Dh2 represents the decision maps.
  [cH2, Dh2]=atb_window(AcH2, BcH2);
  [cV2, Dv2]=atb_window(AcV2, BcV2);
  [cD2, Dd2]=atb_window(AcD2, BcD2);
  [cH1, Dh1]=atb_window(AcH1, BcH1);
  [cV1, Dv1]=atb_window(AcV1, BcV1);
  [cD1, Dd1]=atb_window(AcD1, BcD1);

% The processed approximation and detail coefficients are assembled into C.
  C = fillC( cA2, cH2, cV2, cD2, cH1, cV1, cD1, AS);

% 2-levels reconstruction for the fusion of A and B.
  F=waverec2(C,AS,mywave);
  fusion_scheme = 'Fusion Rule 3 - Window Based Level 2';

% Obtain Quality Index based on Night Vision Image
  [II_qi II_qi_map] = img_qi(double(A), F);

% Obtain Quality Index based on Thermal Image
  [TI_qi TI_qi_map] = img_qi(double(B), F);

    if(II_qi>dbase_II_qi)&&(TI_qi>dbase_TI_qi)
        dbase_II_qi = II_qi;
        dbase_TI_qi = TI_qi;
        dbase_F_img = F;
        dbase_mywave = mywave;
        if (j<=3)
            dbase_contrast = strings1(j);
        else
            dbase_contrast = 0;
        end
        dbase_fusion_scheme = fusion_scheme;
    else
    end
```

```
%---Fusion Rule 1 Level 3------------------------------------------------
%---Wavelet decomposition (3 levels) and extraction of coefficients--------
  [AC,AS]=wavedec2(A_con_enh,3,mywave);
  [BC,BS]=wavedec2(B,3,mywave);

% Extract approximation coefficients of A and B
  AcA3=appcoef2(AC,AS,mywave,3);
  BcA3=appcoef2(BC,BS,mywave,3);
  AcA2=appcoef2(AC,AS,mywave,2);
  BcA2=appcoef2(BC,BS,mywave,2);
  AcA1=appcoef2(AC,AS,mywave,1);
  BcA1=appcoef2(BC,BS,mywave,1);

% Extract detail coefficients  distinguished as horizontal ('h'), vertical
% ('v') and diagonal ('d') details.
  AcH3=detcoef2('h',AC,AS,3);
  AcV3=detcoef2('v',AC,AS,3);
  AcD3=detcoef2('d',AC,AS,3);
  AcH2=detcoef2('h',AC,AS,2);
  AcV2=detcoef2('v',AC,AS,2);
  AcD2=detcoef2('d',AC,AS,2);
  AcH1=detcoef2('h',AC,AS,1);
  AcV1=detcoef2('v',AC,AS,1);
  AcD1=detcoef2('d',AC,AS,1);

  BcH3=detcoef2('h',BC,BS,3);
  BcV3=detcoef2('v',BC,BS,3);
  BcD3=detcoef2('d',BC,BS,3);
  BcH2=detcoef2('h',BC,BS,2);
  BcV2=detcoef2('v',BC,BS,2);
  BcD2=detcoef2('d',BC,BS,2);
  BcH1=detcoef2('h',BC,BS,1);
  BcV1=detcoef2('v',BC,BS,1);
  BcD1=detcoef2('d',BC,BS,1);

% Average approximation coefficients.
  cA3=ave(AcA3, BcA3);
  cA2=ave(AcA2, BcA2);
  cA1=ave(AcA1, BcA1);

% Make new detail coefficients using atb.m. Dh2 represents the decision maps.
  [cH3, Dh3]=atb(AcH3, BcH3);
  [cV3, Dv3]=atb(AcV3, BcV3);
  [cD3, Dd3]=atb(AcD3, BcD3);
  [cH2, Dh2]=atb(AcH2, BcH2);
  [cV2, Dv2]=atb(AcV2, BcV2);
  [cD2, Dd2]=atb(AcD2, BcD2);
  [cH1, Dh1]=atb(AcH1, BcH1);
  [cV1, Dv1]=atb(AcV1, BcV1);
  [cD1, Dd1]=atb(AcD1, BcD1);

% The processed approximation and detail coefficients are assembled into C
  C = fillC3( cA3, cH3, cV3, cD3, cH2, cV2, cD2, cH1, cV1, cD1, AS);
```

```
% 3-levels reconstruction for the fusion of A and B.
 F=waverec2(C,AS,mywave);
 fusion_scheme = 'Fusion Rule 1 Level 3';

% Obtain Quality Index based on Night Vision Image
 [II_qi II_qi_map] = img_qi(double(A), F);

% Obtain Quality Index based on Thermal Image
 [TI_qi TI_qi_map] = img_qi(double(B), F);

  if(II_qi>dbase_II_qi)&&(TI_qi>dbase_TI_qi)
    dbase_II_qi = II_qi;
    dbase_TI_qi = TI_qi;
    dbase_F_img = F;
    dbase_mywave = mywave;
    if (j<=3)
      dbase_contrast = strings1(j);
    else
      dbase_contrast = 0;
    end
    dbase_fusion_scheme = fusion_scheme;
  else
  end

  end                                  % end statement for wavelet family decomposition
end                                  % end statement for image enhancement

% Show fused image.
imshow(dbase_F_img);

% Create overlay for light sources
b_lights=mmcloserec(A,mmsebox(5));
c_lights=mmthreshad(b_lights,200);
d_lights=mmgradm(c_lights);

% Create overlay for heat sources

b_heat=mmcloserec(B,mmsebox(5));
c_heat=mmthreshad(b_heat,200);
d_heat=mmgradm(c_heat);

% Create Null image for overlay use due to limitation on mmshow
[aa1_1,aa1_2] = size(A);
for i=1:aa1_1
  for j= 1:aa1_2
    Anull(i,j)= 0;
  end
end

% Create images with overlays
w_lights = mmshow(dbase_F_img,Anull,Anull,Anull,Anull,c_lights);
w_heat = mmshow(dbase_F_img,c_heat);
hold on
lights = imshow(w_lights);
```

```
set(lights,'AlphaData', 0.5)
heat = imshow(w_heat);
set(heat,'AlphaData', 0.5)
title('Fused Image with light and heat sources ');
dbase_II_qi
dbase_TI_qi
dbase_mywave
dbase_contrast
dbase_fusion_scheme
```

2. WAVELET TRANSFORMATION FUSION APPLICATION WITH GUI INCORPORATED

```
function varargout = application_fusion_GUI(varargin)
% APPLICATION_FUSION_GUI M-file for application_fusion_GUI.fig
%      APPLICATION_FUSION_GUI, by itself, creates a new
APPLICATION_FUSION_GUI or raises the existing
%      singleton*.
%
%      H = APPLICATION_FUSION_GUI returns the handle to a new
APPLICATION_FUSION_GUI or the handle to
%      the existing singleton*.
%
%      APPLICATION_FUSION_GUI('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in APPLICATION_FUSION_GUI.M with the given input
arguments.
%
%      APPLICATION_FUSION_GUI('Property','Value',...) creates a new
APPLICATION_FUSION_GUI or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before application_fusion_GUI_OpeningFunction gets called.
An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to application_fusion_GUI_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help application_fusion_GUI

% Last Modified by GUIDE v2.5 01-Nov-2006 15:42:38

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @application_fusion_GUI_OpeningFcn, ...
                   'gui_OutputFcn',  @application_fusion_GUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before application_fusion_GUI is made visible.
function application_fusion_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to application_fusion_GUI (see VARARGIN)

% Choose default command line output for application_fusion_GUI
handles.output = hObject;

% Update handles structure
% This creates a blank screen in the initialisation of the GUI
guidata(hObject, handles);
    for i=1:40                  % Set size for x dimension
        for j= 1:64             % Set size for y dimension
            Anull(i,j)= 0;      % Set zeros to cover x and y dimension
        end
    end
    axes(handles.axes2);             % Target specific axes
    imshow(Anull);                % Show null image
    axes(handles.axes3);             % Target specific axes
    imshow(Anull);                % Show null image
    axes(handles.axes5);             % Target specific axes
    imshow(Anull);                % Show null image
% UIWAIT makes application_fusion_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = application_fusion_GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes during object creation, after setting all properties.
function uipanel21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NVImage (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function figure1_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.
function figure1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5 as a double

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%        str2double(get(hObject,'String')) returns contents of edit7 as a double
% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%        str2double(get(hObject,'String')) returns contents of edit8 as a double

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%        str2double(get(hObject,'String')) returns contents of edit9 as a double

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton2.
```

```
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

function edit10_Callback(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%        str2double(get(hObject,'String')) returns contents of edit10 as a double

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if          ispc          &&          isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
%        str2double(get(hObject,'String')) returns contents of edit11 as a double

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if          ispc          &&          isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit12 as text
%        str2double(get(hObject,'String')) returns contents of edit12 as a double

% --- Executes during object creation, after setting all properties.
```

```
    function edit12_CreateFcn(hObject, eventdata, handles)
    % hObject    handle to edit12 (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    empty - handles not created until after all CreateFcns called

    % Hint: edit controls usually have a white background on Windows.
    %       See ISPC and COMPUTER.
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

    function edit13_Callback(hObject, eventdata, handles)
    % hObject    handle to edit13 (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)

    % Hints: get(hObject,'String') returns contents of edit13 as text
    %        str2double(get(hObject,'String')) returns contents of edit13 as a double

    % --- Executes during object creation, after setting all properties.
    function edit13_CreateFcn(hObject, eventdata, handles)
    % hObject    handle to edit13 (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    empty - handles not created until after all CreateFcns called

    % Hint: edit controls usually have a white background on Windows.
    %       See ISPC and COMPUTER.
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

    function edit14_Callback(hObject, eventdata, handles)
    % hObject    handle to edit14 (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)

    % Hints: get(hObject,'String') returns contents of edit14 as text
    %        str2double(get(hObject,'String')) returns contents of edit14 as a double

    % --- Executes during object creation, after setting all properties.
    function edit14_CreateFcn(hObject, eventdata, handles)
    % hObject    handle to edit14 (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    empty - handles not created until after all CreateFcns called

    % Hint: edit controls usually have a white background on Windows.
    %       See ISPC and COMPUTER.
    if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

    % --- Executes on button press in loadNVImage.
    function loadNVImage_Callback(hObject, eventdata, handles)
```

```
        % hObject    handle to loadNVImage (see GCBO)
        % eventdata  reserved - to be defined in a future version of MATLAB
        % handles    structure with handles and user data (see GUIDATA)
        [img_nv,path_nv] = uigetfile({'*.bmp';'*.jpg';'*.hdf';'*.png';'*.tif';'*.*'},'Select Night Vision
Image'); % Dialogue screen for image selection

        if isequal(img_nv,0)                              % check image selection, if there is no
selection display msg
          set(handles.commandwindow,'String',...                    % Select output handles and
display message
            'No image selected');
          set(handles.executefusion,'Enable',...               % Set Execute button off
            'off');
        else
          % check image selection, if selected, check file image extension
          [nv_pathstr, nv_name, nv_ext, nv_versn] = fileparts(img_nv)      % supported formats
.bmp, .jpg, .hdf, .png, .tif
          switch lower(nv_ext)
           case '.bmp'                               % Check file extension
              img_nv_ext = nv_ext;                        % Store file extension if file extension
correspond to supported format
           case '.jpg'
              img_nv_ext = nv_ext;
           case '.hdf'
               img_nv_ext = nv_ext;
           case '.png'
               img_nv_ext = nv_ext;
           case '.tif'
               img_nv_ext = nv_ext;
            otherwise
          set(handles.commandwindow,'String',...                      % Set handle and output
message
            'Image type is not supported');
          set(handles.executefusion,'Enable',...               % Set Execute button off
            'off');
          return;
          end
          set(handles.commandwindow,'String',...               % Set handle and output messag
            ['Image Selected ', fullfile(path_nv, img_nv)]);
          % Convert to grayscale if needed
          [A,mapA]=imread(fullfile(path_nv, img_nv));                % read image and stores in
matrix A
          [a1,a2,a3]= size(A);                          % Read size and image info
          if isequal(a3,3)                                  % If image is rgb format convert to
grayscale
            G1 = rgb2gray(A);
            [A,mapA] = gray2ind(G1,256);
          end
          axes(handles.axes2);                           % Set handle for image screen
          imshow(A);                                   % Output image A
          set(handles.text2,'String',...                     % Set handle for text output and output
image file name
            img_nv);
          set(handles.edit9,'String',...                     % Set handle for text output and output
image filename and location
            fullfile(path_nv, img_nv));
          set(handles.edit8,'String',...                    % Set handle for text output and size of
image
             [num2str(a1),' x ',num2str(a2)]);
          set(handles.loadTIImage,'Enable',...                 % Set button for load image B on
             'on');
          set(handles.clearall,'Enable',...                 % Set button for clear images on
             'on');
          setappdata(hObject,'A',A);                        % Output Matrix A
          setappdata(hObject,'mapA',mapA);                     % Output Matrix map A
          setappdata(hObject,'a1',a1);                      % Output Matrix A, x dimension
          setappdata(hObject,'a2',a2);                      % Output Matrix A, y dimension
          setappdata(hObject,'img_nv_ext',img_nv_ext);                % Output Matrix A image
extension
          set(handles.loadNVImage,'Enable',...                  % Set button for load image A off
             'off');
          guidata(hObject, handles);                               % Command needed to output
handles
        end

        %handles.axes2(imshow(A));
        % --- Executes on button press in LoadTIImage.
        function loadTIImage_Callback(hObject, eventdata, handles)
        % hObject    handle to LoadTIImage (see GCBO)
        % eventdata  reserved - to be defined in a future version of MATLAB
        % handles    structure with handles and user data (see GUIDATA)
        % Open Thermal Image
        [img_ti,path_ti] = uigetfile({'*.bmp';'*.jpg';'*.hdf';'*.png';'*.tif';'*.*'},'Select Thermal Image');
% Dialogue screen for image selection

        if isequal(img_ti,0)                              % check image selection, if there is no
selection display msg
          set(handles.commandwindow,'String',...
             'No image selected');
          set(handles.executefusion,'Enable',...
             'off');
        else
          % check image selection, if selected, check file image extension
          [ti_pathstr, ti_name, ti_ext, ti_versn] = fileparts(img_ti)      % supported formats .bmp,
.jpg, .hdf, .png, .tif
          switch lower(ti_ext)
           case '.bmp'
              img_ti_ext = ti_ext;                        % store file extension
           case '.jpg'                                      % Store file extension if file extension
correspond to supported format
              img_ti_ext = ti_ext;
           case '.hdf'
               img_ti_ext = ti_ext;
           case '.png'
               img_ti_ext = ti_ext;
           case '.tif'
               img_ti_ext = ti_ext;
           otherwise
           set(handles.commandwindow,'String',...                     % Set handle and output
message
             'The image type selected is not supported');
           set(handles.executefusion,'Enable',...               % Set Execute button off
             'off');
           return;
          end

          % Convert to grayscale if needed
          [B,mapB]=imread(fullfile(path_ti,img_ti));                    % read image and stores in
matrix B
          [b1,b2,b3]= size(B);                                       % If image is rgb format convert to
grayscale
          if isequal(b3,3)
             G2 = rgb2gray(B);
             [B,mapB] = gray2ind(G2,256);
          end
          A = getappdata(handles.loadNVImage,'A');                      % Get Matrix A and store to
A
          img_nv_ext  =  getappdata(handles.loadNVImage,'img_nv_ext');          %  Get  image
extension of image A
          % Check if image 1 and image 2 dimensions are same size
          if isequal(size(A),size(B))&& isequal(img_nv_ext,img_ti_ext)    % Check image A equal
selected image and file extension corresponds
             axes(handles.axes3);
             imshow(B);                                  % If yes, show image
             set(handles.commandwindow,'String',...                  % Select Handles and output
message
             ['Image Selected ', fullfile(path_ti,img_ti)]);
             set(handles.text3,'String',...                            % Select Handles and output
message
               img_ti);
             set(handles.edit7,'String',...                            % Select Handles and output
message
               fullfile(path_ti, img_ti));
             set(handles.edit5,'String',...                            % Select Handles and output
message
               [num2str(b1),' x ',num2str(b2)]);
             set(handles.executefusion,'Enable',...                 % Set Execute button on
               'on');
             setappdata(hObject,'B',B);                          % Output image B
             setappdata(hObject,'mapB',mapB);                          % Output mapB
             setappdata(hObject,'b1',b1);                         % Output image B, x dimension
             setappdata(hObject,'b2',b2);                         % Output image B, y dimension
             setappdata(hObject,'img_ti_ext',img_ti_ext);                   %  Output  image  B,  file
extenison
             guidata(hObject, handles);
             set(handles.loadTIImage,'Enable',...                   % Set load image B button off
             'off');
             set(handles.clearall,'Enable',...                   % Set clear images button off
             'on');
             set(handles.commandwindow,'String',...                      % Set handles and output
message
               'Click Execute for fusion processing');
          else
             set(handles.commandwindow,'String',...                    % if size not the same, output
ext and set execute button off
             'Thermal image selected do not have the same dimensions or format as Night Vision
image. Please reselect the image.');
             set(handles.executefusion,'Enable',...
              'off');
            return;
          end
        end

        % --- Executes on button press in executefusion.
        function executefusion_Callback(hObject, eventdata, handles)
        % hObject    handle to executefusion (see GCBO)
        % eventdata  reserved - to be defined in a future version of MATLAB
        % handles    structure with handles and user data (see GUIDATA)
        % Initialise quality evaluation parameters
          set(handles.commandwindow,'String',...                    % Set handle and output
message
            'Executing fusion processing....');
          set(handles.executefusion,'Enable',...              % Set execute button off
              'off');
          set(handles.clearall,'Enable',...                   % Set clear images button off
            'off');
          A = getappdata(handles.loadNVImage,'A');                 % Get image A
          mapA = getappdata(handles.loadNVImage,'mapA');               % Get mapA
          a1  =  getappdata(handles.loadNVImage,'a1');                      %  Get  image  A,  x
dimension
          a2  =  getappdata(handles.loadNVImage,'a2');                      %  Get  image  A,  y
dimension
          img_nv_ext = getappdata(handles.loadNVImage,'img_nv_ext');      % Get image A, file
extension
          B = getappdata(handles.loadTIImage,'B');                  % Get image B
          mapB = getappdata(handles.loadTIImage,'mapB');                % Get mapB
          b1 = getappdata(handles.loadTIImage,'b1');                % Get image B, x dimension
          b2 = getappdata(handles.loadTIImage,'b2');                % Get image B, y dimension
          img_ti_ext  =  getappdata(handles.loadTIImage,'img_ti_ext');       %  Get  image  B,  file
extension
        dbase_II_qi = -1;                                     % Set default value quality index for
night vision
        dbase_TI_qi = -1;                                     % Set default value quality index for
thermal image

        %---Start Fusion-----------------------------------------------------------
        % Parameters for image enhancement, define tile sizes and variables
        N = 2;                                            % tile size set at smaller value has better
clarity
        M = 2;                                            % tile size set at smaller value has better
clarity
        data1 = ['uniform    ';'exponential';'rayleigh   '];             % store types of distribution for
adapthisteq function
        celldata1 = cellstr(data1);                         % converts strings to char
        strings1 = char(celldata1);

        % Parameters for wavelet selection
        data2 = ['db2  ';'db5  ';'db7  ';'db9  ';'sym2 ';'sym5 '...
          ;'coif2';'coif5'];                              % store types of distribution for adapthisteq
function
        celldata2 = cellstr(data2);                         % converts strings to char

        % Enhance image via loop selecting a different distribution each time
        for j=1:4
```

```
      if (j<=3)
         A_con_enh = adapthisteq(A,'NumTiles',[N M],'Distribution',strings1(j));
      else

      end
    % Wavelet Selection
      for m=1:8
      mywave = char(celldata2(m));                    % Wavelet selection

    % Wavelet decomposition (2 levels) and extraction of coefficients from
    % reference on wavelet transform fusion is made to fusion.m (DEMO-Wavelets
    % and Image Fusion Centrum voor Wiskunde en Informatica (CWI) -
    % Paul.de.Zeeuw@cwi.nl and Region Based Dec 04 - Khin Choong Chow
      [AC,AS]=wavedec2(A_con_enh,2,mywave);                % Wavelet decomposition
for enhanced image A
      [BC,BS]=wavedec2(B,2,mywave);                      % Wavelet decomposition for
image B

    % Extract approximation coefficients of A and B
      AcA2=appcoef2(AC,AS,mywave,2);
      BcA2=appcoef2(BC,BS,mywave,2);
      AcA1=appcoef2(AC,AS,mywave,1);
      BcA1=appcoef2(BC,BS,mywave,1);

    % Extract detail coefficients  distinguished as horizontal ('h'), vertical
    % ('v') and diagonal ('d') details.
      AcH2=detcoef2('h',AC,AS,2);
      AcV2=detcoef2('v',AC,AS,2);
      AcD2=detcoef2('d',AC,AS,2);
      AcH1=detcoef2('h',AC,AS,1);
      AcV1=detcoef2('v',AC,AS,1);
      AcD1=detcoef2('d',AC,AS,1);

      BcH2=detcoef2('h',BC,BS,2);
      BcV2=detcoef2('v',BC,BS,2);
      BcD2=detcoef2('d',BC,BS,2);
      BcH1=detcoef2('h',BC,BS,1);
      BcV1=detcoef2('v',BC,BS,1);
      BcD1=detcoef2('d',BC,BS,1);

    %---Fusion of approximate image using to region fusion rule----------------
    % Boundary detection. This is done using morphological gradient. Make use
    % of dilation and erosion. This is done for all the three levels.
      AgA1=mmgradm(AcA1);
      AgA2=mmgradm(AcA2);

      BgA1=mmgradm(BcA1);
      BgA2=mmgradm(BcA2);

    % The mmgradm images from the two sources are combined together to form a
    % universal mast for subsequent fusion purposes.
      x1=mmunion(AgA1,BgA1);
      x2=mmunion(AgA2,BgA2);

    % Execute marker-based watershed algorithm
    % Watershed algorithm is performed on this mast to perform a marker-based
```

```
        % region segmentation. Threshold level selected for image 13 for
        % segmentation. A loop is used to limit the number of regions to
        % approximately to 40.
          mh1=MMREGMIN(x1,mmsecross,'dynamics');
          mh2=MMREGMIN(x2,mmsecross,'dynamics');
          nblobs1 = 0;                          % Set regions for level 1 decomposition at 0
          nblobs2 = 0;                          % Set regions for level 2 decomposition at 0
          level1 = 200;                         % Set threshold for level 1 at max value and decrease
one step at a time
          level2 = 200;                         % Set threshold for level 2 at max value and decrease
one step at a time
          while (nblobs1<=38)  % if regions still not less than 38, 38 is select after numerous
experiments to avoid over segementation
            level1 = level1-1; % Decrease threshold level, while loop stops when regions less
than 38
            ws1=mmcwatershed(x1, mmthreshad(mh1, uint8(level1)), mmsecross, 'regions');
            ws1c=mmcwatershed(x1, mmthreshad(mh1, uint8(level1)));
            nblobs1=mmstats(ws1,'max'); % To determine the maximum number of regions for
each subsequent level of decomposition
          end
          while (nblobs2<=36)  % if regions still not less than 36, 36 is select after numerous
experiments to avoid over segementation
            level2=level2-1; % Decrease threshold level, while loop stops when regions less
than 36
            ws2=mmcwatershed(x2, mmthreshad(mh2, uint8(level2)), mmsecross, 'regions');
            ws2c=mmcwatershed(x2, mmthreshad(mh2, uint8(level2)));
            nblobs2=mmstats(ws2,'max'); % To determine the maximum number of regions for
each subsequent level of decomposition
          end
        % Regions are labelled
          L1=mmlabel(ws1c); L2=mmlabel(ws2c);

        % To determine the area of each region
          area1=MMBLOB(ws1,'area');
          area2=MMBLOB(ws2,'area');

        % Make new approximation coefficients using regionfusion.m. DD1 represents
        % the decision maps.
          [Fused_cA1,limit1,DD1] = regionfusion(AcA1,BcA1,ws1,area1,nblobs1);

          [Fused_cA2,limit2,DD2] = regionfusion(AcA2,BcA2,ws2,area2,nblobs2);

        % Make new detail coefficients using atb.m. Dh2 represents the decision maps.
        [cH2, Dh2]=atb(AcH2, BcH2);
        [cV2, Dv2]=atb(AcV2, BcV2);
        [cD2, Dd2]=atb(AcD2, BcD2);
        [cH1, Dh1]=atb(AcH1, BcH1);
        [cV1, Dv1]=atb(AcV1, BcV1);
        [cD1, Dd1]=atb(AcD1, BcD1);

        % The processed approximation and detail coefficients are assembled into C.
        C = fillC( Fused_cA2, cH2, cV2, cD2, cH1, cV1, cD1, AS);

        % 2-levels reconstruction for the fusion of A and B.
        F=waverec2(C,AS,mywave);
        fusion_scheme = 'Region Based Fusion'; % Labelled as region based fusion
        % Obtain Quality Index based on Night Vision Image
        [II_qi II_qi_map] = img_qi(double(A), F);
        % Obtain Quality Index based on Thermal Image
        [TI_qi TI_qi_map] = img_qi(double(B), F);
        set(handles.commandwindow,'String',...      % Output busy message
          'Busy....');
          if(II_qi>dbase_II_qi)&&(TI_qi>dbase_TI_qi)% Check quality indexes for night vision
and thermal images if both greater than existing ones, replace
            dbase_II_qi = II_qi;              % Store new quality index
            dbase_TI_qi = TI_qi;
            dbase_F_img = F;                  % Store corresponding fusion scheme
            dbase_mywave = mywave;            % Store corresponding wavelet family
            if (j<=3)                         % Store corresponding contrast distribution
              dbase_contrast = strings1(j);
            else
              dbase_contrast = 0;             % Else set to defualt contrast distribution
            end
            dbase_fusion_scheme = fusion_scheme;
          else
          end                                               % end for if statement

        %---Fusion Rule 1 Weighted Average - Level 2-------------------------------
        % Average approximation coefficients.
         cA2=ave(AcA2, BcA2);
         cA1=ave(AcA1, BcA1);

        % Make new detail coefficients using atb.m. Dh2 represents the decision maps.
         [cH2, Dh2]=atb(AcH2, BcH2);
         [cV2, Dv2]=atb(AcV2, BcV2);
         [cD2, Dd2]=atb(AcD2, BcD2);
         [cH1, Dh1]=atb(AcH1, BcH1);
         [cV1, Dv1]=atb(AcV1, BcV1);
         [cD1, Dd1]=atb(AcD1, BcD1);

        % The processed approximation and detail coefficients are assembled into C.
         C = fillC( cA2, cH2, cV2, cD2, cH1, cV1, cD1, AS);

        % 2-levels reconstruction for the fusion of A and B.
         F=waverec2(C,AS,mywave);
         fusion_scheme = 'Fusion Rule 1 Level 2'; % Labelled as fusion rule 1 level 2
        % Obtain Quality Index based on Night Vision Image
         [II_qi II_qi_map] = img_qi(double(A), F);

        % Obtain Quality Index based on Thermal Image
         [TI_qi TI_qi_map] = img_qi(double(B), F);

          if(II_qi>dbase_II_qi)&&(TI_qi>dbase_TI_qi)    % Similar commands to store quality
indexes and infor on fused image described in region based
            dbase_II_qi = II_qi;
            dbase_TI_qi = TI_qi;
            dbase_F_img = F;
            dbase_mywave = mywave;
            if (j<=3)
              dbase_contrast = strings1(j);
            else
```

```
             dbase_contrast = 0;
           end
           dbase_fusion_scheme = fusion_scheme;
         else
         end                                                          % end for if statement
       %---Fusion Rule 2 Activity Based- Level 2-----------------------------------
       % Average approximation coefficients.
        cA2=ave(AcA2, BcA2);
        cA1=ave(AcA1, BcA1);

       % Make new detail coefficients using atb.m. Dh2 represents the decision maps.
        [cH2, Dh2]=atb_activity(AcH2, BcH2);
        [cV2, Dv2]=atb_activity(AcV2, BcV2);
        [cD2, Dd2]=atb_activity(AcD2, BcD2);
        [cH1, Dh1]=atb_activity(AcH1, BcH1);
        [cV1, Dv1]=atb_activity(AcV1, BcV1);
        [cD1, Dd1]=atb_activity(AcD1, BcD1);

       % The processed approximation and detail coefficients are assembled into C.
        C = fillC( cA2, cH2, cV2, cD2, cH1, cV1, cD1, AS);

       % 2-levels reconstruction for the fusion of A and B.
        F=waverec2(C,AS,mywave);
        fusion_scheme = 'Fusion Rule 2 - Activity Based Level 2'; % Labelled as fusion rule 2

       % Obtain Quality Index based on Night Vision Image
        [II_qi II_qi_map] = img_qi(double(A), F);

       % Obtain Quality Index based on Thermal Image
        [TI_qi TI_qi_map] = img_qi(double(B), F);

         if(II_qi>dbase_II_qi)&&(TI_qi>dbase_TI_qi)  %  Similar  commands  to  store  quality
indexes and infor on fused image described in region based
           dbase_II_qi = II_qi;
           dbase_TI_qi = TI_qi;
           dbase_F_img = F;
           dbase_mywave = mywave;
           if (j<=3)
             dbase_contrast = strings1(j);
           else
             dbase_contrast = 0;
           end
           dbase_fusion_scheme = fusion_scheme;
         else
         end
       %---Fusion Rule 2 Window Based- Level 2-----------------------------------
       % Average approximation coefficients.
        cA2=ave(AcA2, BcA2);
        cA1=ave(AcA1, BcA1);

       % Make new detail coefficients using atb.m. Dh2 represents the decision maps.
        [cH2, Dh2]=atb_window(AcH2, BcH2);
        [cV2, Dv2]=atb_window(AcV2, BcV2);
        [cD2, Dd2]=atb_window(AcD2, BcD2);
        [cH1, Dh1]=atb_window(AcH1, BcH1);
        [cV1, Dv1]=atb_window(AcV1, BcV1);
```

```
 [cD1, Dd1]=atb_window(AcD1, BcD1);

% The processed approximation and detail coefficients are assembled into C.
 C = fillC( cA2, cH2, cV2, cD2, cH1, cV1, cD1, AS);

% 2-levels reconstruction for the fusion of A and B.
 F=waverec2(C,AS,mywave);
 fusion_scheme = 'Fusion Rule 3 - Window Based Level 2'; % Labelled as fusion rule 3

% Obtain Quality Index based on Night Vision Image
 [II_qi II_qi_map] = img_qi(double(A), F);

% Obtain Quality Index based on Thermal Image
 [TI_qi TI_qi_map] = img_qi(double(B), F);

  if(II_qi>dbase_II_qi)&&(TI_qi>dbase_TI_qi) % Similar commands to store quality indexes and infor on fused image described in region based
    dbase_II_qi = II_qi;
    dbase_TI_qi = TI_qi;
    dbase_F_img = F;
    dbase_mywave = mywave;
    if (j<=3)
      dbase_contrast = strings1(j);
    else
      dbase_contrast = 0;
    end
    dbase_fusion_scheme = fusion_scheme;
  else
  end
%---Fusion Rule 1 Level 3-------------------------------------------------
%---Wavelet decomposition (3 levels) and extraction of coefficients--------
 [AC,AS]=wavedec2(A_con_enh,3,mywave);
 [BC,BS]=wavedec2(B,3,mywave);

% Extract approximation coefficients of A and B
 AcA3=appcoef2(AC,AS,mywave,3);
 BcA3=appcoef2(BC,BS,mywave,3);
 AcA2=appcoef2(AC,AS,mywave,2);
 BcA2=appcoef2(BC,BS,mywave,2);
 AcA1=appcoef2(AC,AS,mywave,1);
 BcA1=appcoef2(BC,BS,mywave,1);

% Extract detail coefficients  distinguished as horizontal ('h'), vertical
% ('v') and diagonal ('d') details.
 AcH3=detcoef2('h',AC,AS,3);
 AcV3=detcoef2('v',AC,AS,3);
 AcD3=detcoef2('d',AC,AS,3);
 AcH2=detcoef2('h',AC,AS,2);
 AcV2=detcoef2('v',AC,AS,2);
 AcD2=detcoef2('d',AC,AS,2);
 AcH1=detcoef2('h',AC,AS,1);
 AcV1=detcoef2('v',AC,AS,1);
 AcD1=detcoef2('d',AC,AS,1);

 BcH3=detcoef2('h',BC,BS,3);
 BcV3=detcoef2('v',BC,BS,3);
```

```
     BcD3=detcoef2('d',BC,BS,3);
     BcH2=detcoef2('h',BC,BS,2);
     BcV2=detcoef2('v',BC,BS,2);
     BcD2=detcoef2('d',BC,BS,2);
     BcH1=detcoef2('h',BC,BS,1);
     BcV1=detcoef2('v',BC,BS,1);
     BcD1=detcoef2('d',BC,BS,1);

    % Average approximation coefficients.
     cA3=ave(AcA3, BcA3);
     cA2=ave(AcA2, BcA2);
     cA1=ave(AcA1, BcA1);

    % Make new detail coefficients using atb.m. Dh2 represents the decision maps.
     [cH3, Dh3]=atb(AcH3, BcH3);
     [cV3, Dv3]=atb(AcV3, BcV3);
     [cD3, Dd3]=atb(AcD3, BcD3);
     [cH2, Dh2]=atb(AcH2, BcH2);
     [cV2, Dv2]=atb(AcV2, BcV2);
     [cD2, Dd2]=atb(AcD2, BcD2);
     [cH1, Dh1]=atb(AcH1, BcH1);
     [cV1, Dv1]=atb(AcV1, BcV1);
     [cD1, Dd1]=atb(AcD1, BcD1);

    % The processed approximation and detail coefficients are assembled into C
     C = fillC3( cA3, cH3, cV3, cD3, cH2, cV2, cD2, cH1, cV1, cD1, AS);

    % 3-levels reconstruction for the fusion of A and B.
     F=waverec2(C,AS,mywave);
     fusion_scheme = 'Fusion Rule 1 Level 3'; % Labelled as fusion rule 1

    % Obtain Quality Index based on Night Vision Image
     [II_qi II_qi_map] = img_qi(double(A), F);

    % Obtain Quality Index based on Thermal Image
     [TI_qi TI_qi_map] = img_qi(double(B), F);

      if(II_qi>dbase_II_qi)&&(TI_qi>dbase_TI_qi) % Similar commands to store quality indexes and infor on fused image described in region based
        dbase_II_qi = II_qi;
        dbase_TI_qi = TI_qi;
        dbase_F_img = F;
        dbase_mywave = mywave;
        if (j<=3)
          dbase_contrast = strings1(j);
        else
          dbase_contrast = 0;
        end
        dbase_fusion_scheme = fusion_scheme;
      else
      end

      end                                    % end statement for wavelet family decomposition
    end                                 % end statement for image enhancement
```

```
        % Show fused image.
          dbase_F_img_F  = mmshow(dbase_F_img);   % Show fused image using SDC toolbox
          axes(handles.axes5);                   % Select handles
          imshow(dbase_F_img_F);                   % Show fused image
          set(handles.commandwindow,'String',...
            'Image Fusion Complete');

        % Create overlay for light sources
        b_lights=mmcloserec(A,mmsebox(5));
        c_lights=mmthreshad(b_lights,210);
        d_lights=mmgradm(c_lights);

        % Create overlay for heat sources

        b_heat=mmcloserec(B,mmsebox(5));
        c_heat=mmthreshad(b_heat,210);
        d_heat=mmgradm(c_heat);

        % Create Null image for overlay use due to limitation on mmshow
        [aa1_1,aa1_2] = size(A);
        for i=1:aa1_1           %create null overlays based on image A or B size
          for j= 1:aa1_2
            Anull(i,j)= 0;
          end
        end

        % Create images with overlays
        w_lights = mmshow(dbase_F_img,Anull,Anull,Anull,Anull,c_lights);    % Show night vision
overlay
        w_heat = mmshow(dbase_F_img,c_heat);                               % Show thermal image
overlay
        dbase_F_img_OL  =  mmshow(dbase_F_img,c_heat,Anull,Anull,Anull,c_lights);  %  Show
fused inmage with overlays
        set(handles.edit10,'String',...
          dbase_mywave);                                   % Set handles and output message
        set(handles.edit11,'String',...
          dbase_contrast);                                 % Set handles and output message
        set(handles.edit12,'String',...
              num2str(dbase_II_qi));                         % Set handles and output message
        set(handles.edit13,'String',...
          num2str(dbase_TI_qi));                              % Set handles and output message
        set(handles.edit14,'String',...
          dbase_fusion_scheme);                                % Set handles and output message
        set(handles.clearall,'Enable',...
          'on');                                        % Set clear images button on
        setappdata(hObject,'A_con_enh',A_con_enh);                      % Set handles and output
data
        setappdata(hObject,'dbase_F_img_F',dbase_F_img_F);                  % Set handles and
output data
        setappdata(hObject,'dbase_F_img_OL',dbase_F_img_OL);                % Set handles and
output data
        guidata(hObject, handles);                            % Needed to output data
        if isequal(dbase_fusion_scheme,'Region Based Fusion')               % Set region based
related button on only if region based is selected
          setappdata(hObject,'AgA1',AgA1);                        % Output all related data
          setappdata(hObject,'BgA1',BgA1);
```

```
    set(handles.NVRegions,'Enable',...
    'on');
    set(handles.TIRegions,'Enable',...
    'on');
    setappdata(hObject,'AcA1',AcA1);
    setappdata(hObject,'AcH1',AcH1);
    setappdata(hObject,'AcV1',AcV1);
    setappdata(hObject,'AcD1',AcD1);
    setappdata(hObject,'AcA2',AcA2);
    setappdata(hObject,'AcH2',AcH2);
    setappdata(hObject,'AcV2',AcV2);
    setappdata(hObject,'AcD2',AcD2);
    setappdata(hObject,'BcA1',BcA1);
    setappdata(hObject,'BcH1',BcH1);
    setappdata(hObject,'BcV1',BcV1);
    setappdata(hObject,'BcD1',BcD1);
    setappdata(hObject,'BcA2',BcA2);
    setappdata(hObject,'BcH2',BcH2);
    setappdata(hObject,'BcV2',BcV2);
    setappdata(hObject,'BcD2',BcD2);
    setappdata(hObject,'w_lights',w_lights);
    setappdata(hObject,'w_heat',w_heat);
    set(handles.clearall,'Enable',...
        'on');
    set(handles.NVLevel1,'Enable',...
        'on');
    set(handles.TILevel1,'Enable',...
        'on');
    set(handles.NVLevel2,'Enable',...
        'on');
    set(handles.TILevel2,'Enable',...
        'on');
    set(handles.savefused,'Enable',...
        'on');
    set(handles.executefusion,'Enable',...
        'off');
    set(handles.showNV,'Enable',...
        'on');
    guidata(hObject, handles);
else
    setappdata(hObject,'AcA1',AcA1);
    setappdata(hObject,'AcH1',AcH1);
    setappdata(hObject,'AcV1',AcV1);
    setappdata(hObject,'AcD1',AcD1);
    setappdata(hObject,'AcA2',AcA2);
    setappdata(hObject,'AcH2',AcH2);
    setappdata(hObject,'AcV2',AcV2);
    setappdata(hObject,'AcD2',AcD2);
    setappdata(hObject,'BcA1',BcA1);
    setappdata(hObject,'BcH1',BcH1);
    setappdata(hObject,'BcV1',BcV1);
    setappdata(hObject,'BcD1',BcD1);
    setappdata(hObject,'BcA2',BcA2);
    setappdata(hObject,'BcH2',BcH2);
    setappdata(hObject,'BcV2',BcV2);
    setappdata(hObject,'BcD2',BcD2);
    setappdata(hObject,'w_lights',w_lights);
    setappdata(hObject,'w_heat',w_heat);
    set(handles.clearall,'Enable',...
        'on');
    set(handles.NVLevel1,'Enable',...
        'on');
    set(handles.TILevel1,'Enable',...
        'on');
    set(handles.NVLevel2,'Enable',...
        'on');
    set(handles.TILevel2,'Enable',...
        'on');
    set(handles.savefused,'Enable',...
        'on');
    set(handles.showNV,'Enable',...
        'on');
    set(handles.executefusion,'Enable',...
        'off');
    guidata(hObject, handles);
end

% Button to display coefficients at level 1 of decomposition for night
% vision
function NVLevel1_Callback(hObject, eventdata, handles)
AcA1 = getappdata(handles.executefusion,'AcA1');        % Get data and store
AcH1 = getappdata(handles.executefusion,'AcH1');        % Get data and store
AcV1 = getappdata(handles.executefusion,'AcV1');        % Get data and store
AcD1 = getappdata(handles.executefusion,'AcD1');        % Get data and store
mapA = getappdata(handles.loadNVImage,'mapA');          % Get data and store
figure
title('Level 1 approximation and detail coefficients for Night Vision');
subplot(2,2,3)
image(AcA1)
title('Approximation A1')
colormap(mapA)
subplot(2,2,4)
image(AcH1);
colormap(mapA);
title('Horizontal Detail H1')
subplot(2,2,1)
image(AcV1);
colormap(mapA);
title('Vertical Detail V1')
subplot(2,2,2)
image(AcD1);
title('Diagonal Detail D1')

% Button to display coefficients at level 2 of decomposition for night
% vision
function NVLevel2_Callback(hObject, eventdata, handles)
AcA2 = getappdata(handles.executefusion,'AcA2');    % Get data and store
AcH2 = getappdata(handles.executefusion,'AcH2');    % Get data and store
AcV2 = getappdata(handles.executefusion,'AcV2');    % Get data and store
AcD2 = getappdata(handles.executefusion,'AcD2');    % Get data and store
mapA = getappdata(handles.loadNVImage,'mapA');      % Get data and store
figure  % plot
```

```
title('Level 2 approximation and detail coefficients for Night Vision');
subplot(2,2,3)
image(AcA2)
title('Approximation A2')
colormap(mapA)
subplot(2,2,4)
image(AcH2);
colormap(mapA);
title('Horizontal Detail H2')
subplot(2,2,1)
image(AcV2);
colormap(mapA);
title('Vertical Detail V2')
subplot(2,2,2)
image(AcD2);
title('Diagonal Detail D2')

% Button to display coefficients at level 1 of decomposition for thermal
% image
function TILevel1_Callback(hObject, eventdata, handles)
% --- Executes on button press in NVLevel2.
BcA1 = getappdata(handles.executefusion,'BcA1');    % Get data and store
BcH1 = getappdata(handles.executefusion,'BcH1');    % Get data and store
BcV1 = getappdata(handles.executefusion,'BcV1');    % Get data and store
BcD1 = getappdata(handles.executefusion,'BcD1');    % Get data and store
mapB = getappdata(handles.loadTIImage,'mapB');      % Get data and store
figure % Plot
title('Level 2 approximation and detail coefficients for Thermal Image');
subplot(2,2,3)
image(BcA1)
title('Approximation A1')
colormap(mapB)
subplot(2,2,4)
image(BcH1);
colormap(mapB);
title('Horizontal Detail H1')
subplot(2,2,1)
image(BcV1);
colormap(mapB);
title('Vertical Detail V1')
subplot(2,2,2)
image(BcD1);
title('Diagonal Detail D1')

% Button to display coefficients at level 2 of decomposition for thermal
% image
function TILevel2_Callback(hObject, eventdata, handles)
BcA2 = getappdata(handles.executefusion,'BcA2');    % Get data and store
BcH2 = getappdata(handles.executefusion,'BcH2');    % Get data and store
BcV2 = getappdata(handles.executefusion,'BcV2');    % Get data and store
BcD2 = getappdata(handles.executefusion,'BcD2');    % Get data and store
mapB = getappdata(handles.loadTIImage,'mapB');
figure %plot
title('Level 1 approximation and detail coefficients for Thermal Image');
subplot(2,2,3)
image(BcA2)
```

```
title('Approximation A2')
colormap(mapB)
subplot(2,2,4)
image(BcH2);
colormap(mapB);
title('Horizontal Detail H2')
subplot(2,2,1)
image(BcV2);
colormap(mapB);
title('Vertical Detail V2')
subplot(2,2,2)
image(BcD2);
title('Diagonal Detail D2')

% Button to display segementation at level 1 of decomposition for night
% vision segmenation
function NVRegions_Callback(hObject, eventdata, handles)
AgA1 = getappdata(handles.executefusion,'AgA1');    % Get data and store
figure
title('Night Vision Region Segmenation');
imshow(AgA1);

% Button to display segementation at level 1 of decomposition for thermal
% image segmentation
function TIRegions_Callback(hObject, eventdata, handles)
BgA1 = getappdata(handles.executefusion,'BgA1');  % Get data and store
figure
title('Thermal Image Region Segmenation');
imshow(BgA1);

% Button to clear all images and set buttons
function clearall_Callback(hObject, eventdata, handles)
   set(handles.clearall,'Enable',...
      'off');
   set(handles.NVLevel1,'Enable',...
      'off');
   set(handles.TILevel1,'Enable',...
      'off');
   set(handles.NVLevel2,'Enable',...
      'off');
   set(handles.TILevel2,'Enable',...
      'off');
   set(handles.NVRegions,'Enable',...
      'off');
   set(handles.TIRegions,'Enable',...
      'off');
   set(handles.savefused,'Enable',...
      'off');
   set(handles.executefusion,'Enable',...
      'off');
   set(handles.loadNVImage,'Enable',...
      'on');
   set(handles.loadTIImage,'Enable',...
      'off');
   set(handles.showNV,'Enable',...
      'off');
```

```
    %create null for blank screen
    for i=1:270
        for j= 1:360
            Anull(i,j)= 0;
        end
    end
    axes(handles.axes2);
    imshow(Anull);
    axes(handles.axes3);
    imshow(Anull);
    axes(handles.axes5);
    imshow(Anull);
    set(handles.commandwindow,'String',...
        'No image selected');
    set(handles.text3,'String',...
        ' ');
    set(handles.edit7,'String',...
        ' ');
    set(handles.edit5,'String',...
        ' ');
    set(handles.text2,'String',...
        ' ');
    set(handles.edit9,'String',...
        ' ');
    set(handles.edit8,'String',...
        ' ');
    set(handles.edit10,'String',...
        ' ');
    set(handles.edit11,'String',...
        ' ');
    set(handles.edit12,'String',...
        ' ');
    set(handles.edit13,'String',...
        ' ');
    set(handles.edit14,'String',...
        ' ');

function commandwindow_Callback(hObject, eventdata, handles)

function edit15_Callback(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit15 as text
%        str2double(get(hObject,'String')) returns contents of edit15 as a double
% --- Executes on button press in savefused.

% Button to save fused image
function savefused_Callback(hObject, eventdata, handles)
% hObject    handle to savefused (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
dbase_F_img_OL = getappdata(handles.executefusion,'dbase_F_img_OL');
figure;
imshow(dbase_F_img_OL);
```

```
% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% Button to display both overlay with fused image
function bothoverlay_Callback(hObject, eventdata, handles)
dbase_F_img_OL = getappdata(handles.executefusion,'dbase_F_img_OL');
axes(handles.axes5);
imshow(dbase_F_img_OL);

% Button to display night vision overlay with fused image
function NVoverlay_Callback(hObject, eventdata, handles)
w_lights = getappdata(handles.executefusion,'w_lights');
axes(handles.axes5);
imshow(w_lights);

% Button to display thermal image overlay with fused image
function TIoverlay_Callback(hObject, eventdata, handles)

w_heat = getappdata(handles.executefusion,'w_heat');
axes(handles.axes5);
imshow(w_heat);

function axes5_CreateFcn(hObject, eventdata, handles)

function axes3_CreateFcn(hObject, eventdata, handles)

function axes2_CreateFcn(hObject, eventdata, handles)

function text2_Callback(hObject, eventdata, handles)

function text3_Callback(hObject, eventdata, handles)

function NVImage_CreateFcn(hObject, eventdata, handles)

function TIImage_CreateFcn(hObject, eventdata, handles)

function fusedimage_CreateFcn(hObject, eventdata, handles)

function uipanel16_CreateFcn(hObject, eventdata, handles)

function uipanel17_CreateFcn(hObject, eventdata, handles)

function uipanel15_CreateFcn(hObject, eventdata, handles)

function uipanel18_CreateFcn(hObject, eventdata, handles)
function uipanel20_CreateFcn(hObject, eventdata, handles)

% Button to display only fused image
function none_Callback(hObject, eventdata, handles)
dbase_F_img_F = getappdata(handles.executefusion,'dbase_F_img_F');
axes(handles.axes5);
imshow(dbase_F_img_F);

% Button to contrast enhance night vision image
function showNV_Callback(hObject, eventdata, handles)
A_con_enh = getappdata(handles.executefusion,'A_con_enh');
figure;
title('Contrast enhanced Night Vision Image');
imshow(A_con_enh);
```

APPENDIX B: DETAIL GUI APPLICATION FLOWCHART

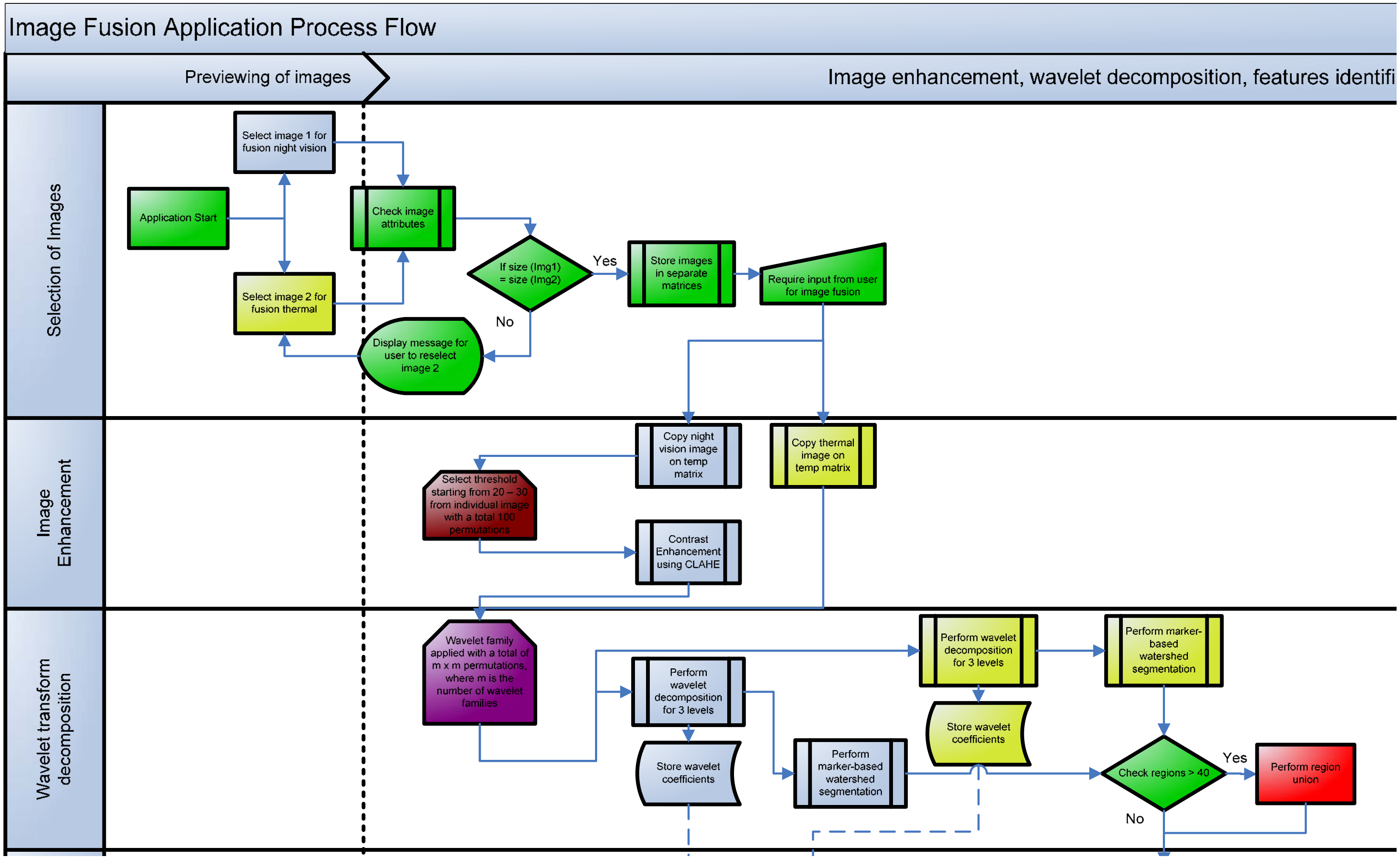
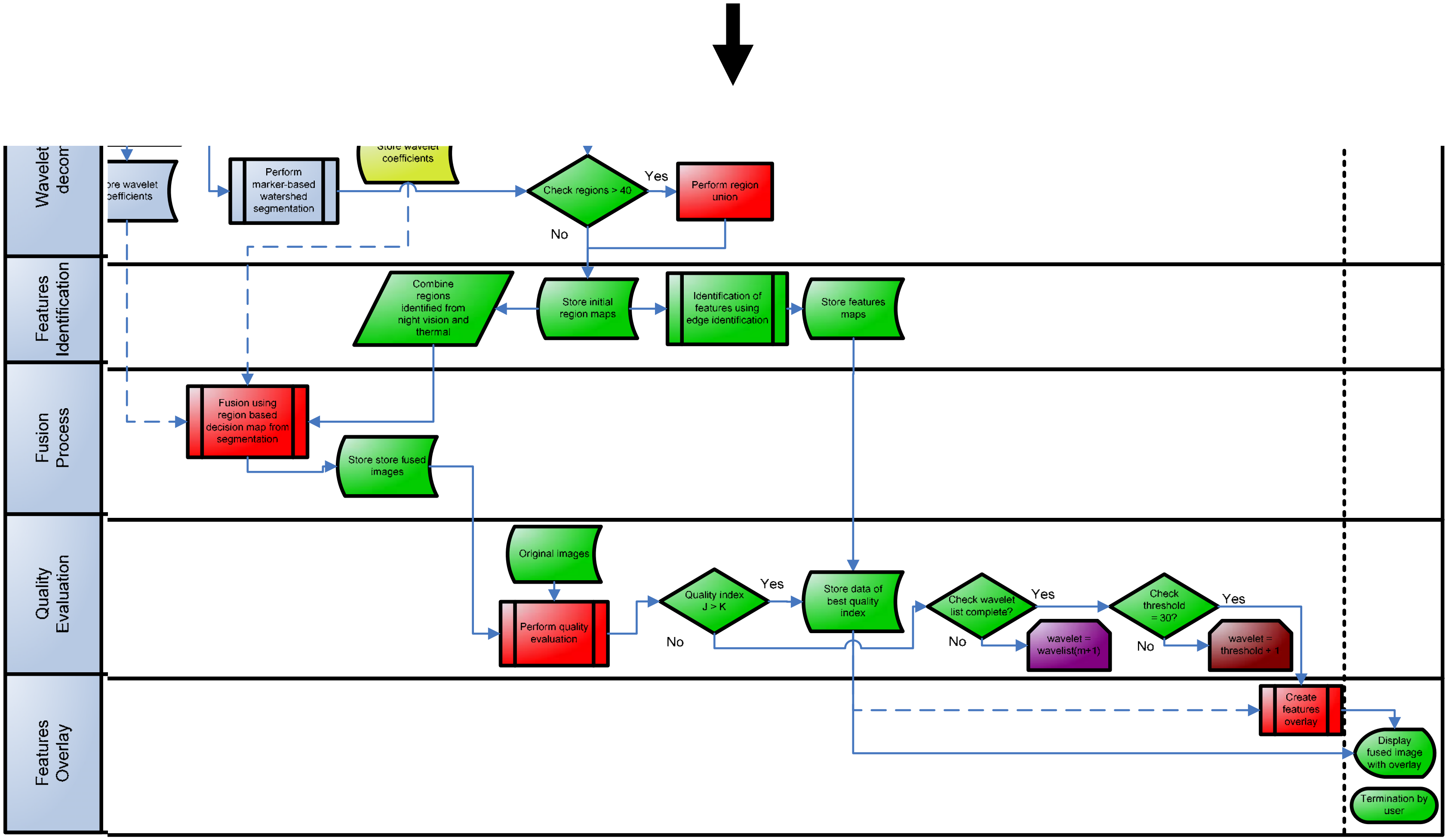


Image Fusion Application Process Flow
Previewing of images
Image enhancement, wavelet decomposition, features identifi
Selection of Images
Image Enhancement
Wavelet transform decomposition
Select image 1 for fusion night vision
Application Start
Check image attributes
Select image 2 for fusion thermal
If size (Img1) = size (Img2)
Yes
No
Store images in separate matrices
Require input from user for image fusion
Display message for user to reselect image 2
Copy night vision image on temp matrix
Copy thermal image on temp matrix
Select threshold starting from 20 – 30 from individual image with a total 100 permutations
Contrast Enhancement using CLAHE
Wavelet family applied with a total of m x m permutations, where m is the number of wavelet families
Perform wavelet decomposition for 3 levels
Store wavelet coefficients
Perform wavelet decomposition for 3 levels
Store wavelet coefficients
Perform marker-based watershed segmentation
Perform marker-based watershed segmentation
Check regions > 40
Yes
No
Perform region union

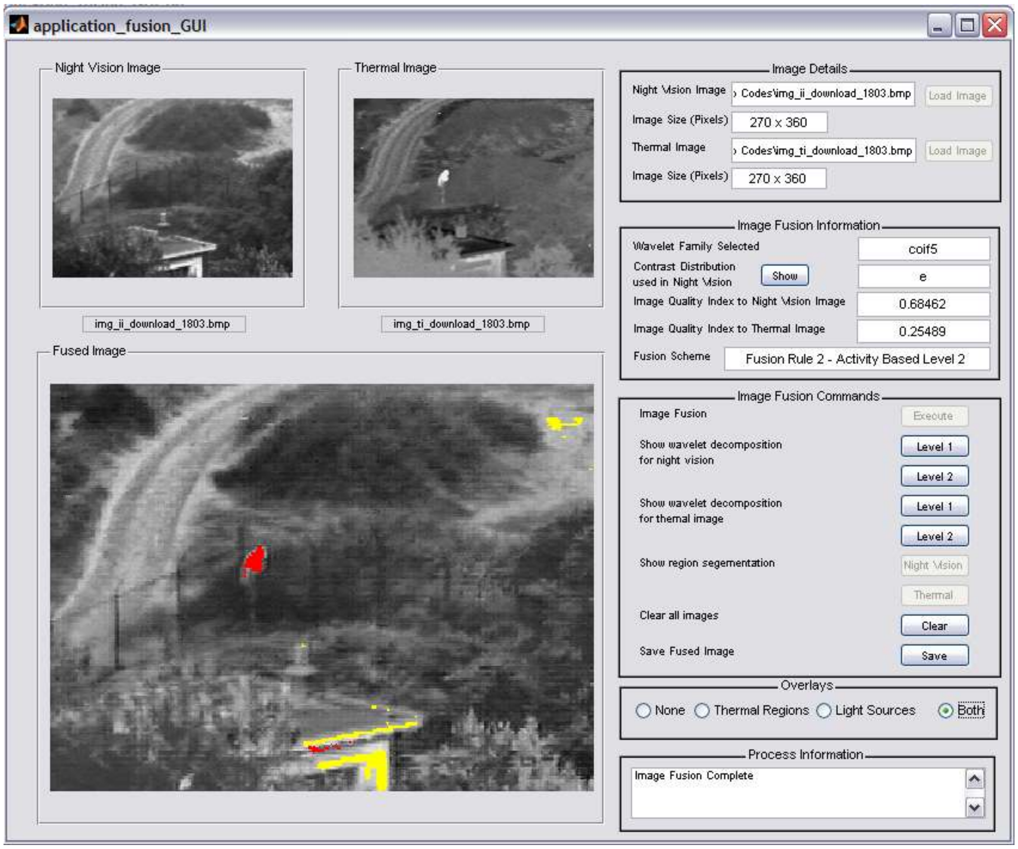

Wavelet
decom
Features
Identification
Fusion
Process
Quality
Evaluation
Features
Overlay
ore wavelet
oefficients
Perform
marker-based
watershed
segmentation
Store wavelet
coefficients
Check regions > 40
Yes
No
Perform region
union
Combine
regions
identified from
night vision and
thermal
Store initial
region maps
Identification of
features using
edge identification
Store features
maps
Fusion using
region based
decision map from
segmentation
Store store fused
images
Original images
Perform quality
evaluation
Quality index
J > K
Yes
No
Store data of
best quality
index
Check wavelet
list complete?
Yes
No
wavelet =
wavelist(m+1)
Check
threshold
= 30?
Yes
No
wavelet =
threshold + 1
Create
features
overlay
Display
fused image
with overlay
Termination by
user

APPENDIX C. GUI EXPERIMENT RESULTS

This Appendix contains the experiment results of one of three images applied to the GUI application:

A. Image 1803.bmp

1. Output Screen



application_fusion_GUI

Night Vision Image — img_ii_download_1803.bmp

Thermal Image — img_ti_download_1803.bmp

Fused Image

Image Details

Night Vision Image: › Codes\img_ii_download_1803.bmp [Load Image]

Image Size (Pixels): 270 x 360

Thermal Image: › Codes\img_ti_download_1803.bmp [Load Image]

Image Size (Pixels): 270 x 360

Image Fusion Information

Wavelet Family Selected: coif5

Contrast Distribution used in Night Vision [Show]: e

Image Quality Index to Night Vision Image: 0.68462

Image Quality Index to Thermal Image: 0.25489

Fusion Scheme: Fusion Rule 2 - Activity Based Level 2

Image Fusion Commands

Image Fusion [Execute]

Show wavelet decomposition for night vision [Level 1] [Level 2]

Show wavelet decomposition for thermal image [Level 1] [Level 2]

Show region segementation [Night Vision] [Thermal]

Clear all images [Clear]

Save Fused Image [Save]

Overlays

○ None ○ Thermal Regions ○ Light Sources ● Both

Process Information

Image Fusion Complete

2. Data

Image	Fusion Scheme	NV Quality Index	TI Quality Index	Wavelet Family	CLAHE Type	Rank based on TI Quality Index	Rank based on TI Quality Index
1803	Fusion Rule 2	0.68666	0.24729	'sym5'	'e'	1	91
1803	Fusion Rule 2	0.68462	0.25489	'coif5'	'e'	2	76
1803	Fusion Rule 2	0.68148	0.25083	'db5'	'e'	3	84
1803	Fusion Rule 2	0.6812	0.25432	'coif2'	'e'	4	78
1803	Fusion Rule 2	0.68009	0.31119	'sym5'	'r'	5	48
1803	Fusion Rule 2	0.67977	0.25199	'db9'	'e'	6	82
1803	Fusion Rule 2	0.67967	0.25454	'db7'	'e'	7	77
1803	Fusion Rule 2	0.67592	0.32003	'coif5'	'r'	8	41
1803	Fusion Rule 2	0.67394	0.31533	'db5'	'r'	9	47

1803	Region Based	0.67295	0.30555	'db9'	'r'	10	59
1803	Fusion Rule 1	0.67295	0.30555	'db9'	'r'	11	60
1803	Fusion Rule 2	0.67283	0.24113	'sym5'	'u'	12	96
1803	Fusion Rule 2	0.6727	0.31682	'db9'	'r'	13	46
1803	Fusion Rule 2	0.6726	0.31945	'coif2'	'r'	14	43
1803	Fusion Rule 1	0.67228	0.31023	'coif5'	'r'	15	49
1803	Region Based	0.67206	0.308	'db7'	'r'	16	55
1803	Fusion Rule 1	0.67206	0.308	'db7'	'r'	17	56
1803	Fusion Rule 2	0.67157	0.31993	'db7'	'r'	18	42
1803	Region Based	0.67126	0.30919	'coif5'	'r'	19	52
1803	Fusion Rule 2	0.67072	0.24867	'coif5'	'u'	20	86
1803	Fusion Rule 1	0.67059	0.30783	'coif2'	'r'	21	57
1803	Region Based	0.67006	0.30932	'sym5'	'r'	22	50
1803	Fusion Rule 1	0.67006	0.30932	'sym5'	'r'	23	51
1803	Region Based	0.66966	0.30685	'coif2'	'r'	24	58
1803	Region Based	0.66899	0.30893	'db5'	'r'	25	53
1803	Fusion Rule 1	0.66899	0.30893	'db5'	'r'	26	54
1803	Fusion Rule 2	0.66857	0.25337	'db2'	'e'	27	79
1803	Fusion Rule 2	0.66857	0.25337	'sym2'	'e'	28	80
1803	Fusion Rule 2	0.66761	0.24483	'db5'	'u'	29	95
1803	Fusion Rule 2	0.66728	0.24795	'coif2'	'u'	30	88
1803	Fusion Rule 2 WB	0.66705	0.27701	'coif5'	'r'	31	67
1803	Fusion Rule 2 WB	0.66677	0.26872	'db5'	'r'	32	70
1803	Fusion Rule 2	0.6663	0.24602	'db9'	'u'	33	94
1803	Fusion Rule 2	0.66615	0.24833	'db7'	'u'	34	87
1803	Region Based	0.66442	0.29772	'db2'	'r'	35	61
1803	Region Based	0.66442	0.29772	'sym2'	'r'	36	62
1803	Fusion Rule 1	0.66442	0.29772	'db2'	'r'	37	63
1803	Fusion Rule 1	0.66442	0.29772	'sym2'	'r'	38	64
1803	Fusion Rule 2 WB	0.66366	0.27395	'coif2'	'r'	39	69
1803	Fusion Rule 2 WB	0.66332	0.27489	'db9'	'r'	40	68
1803	Fusion Rule 2 WB	0.66309	0.20926	'coif5'	'e'	41	130
1803	Fusion Rule 2	0.66292	0.31762	'db2'	'r'	42	44
1803	Fusion Rule 2	0.66292	0.31762	'sym2'	'r'	43	45
1803	Fusion Rule 2 WB	0.66213	0.27813	'db7'	'r'	44	65
1803	Fusion Rule 2 WB	0.66207	0.27777	'sym5'	'r'	45	66
1803	Fusion Rule 1	0.66091	0.23132	'db9'	'e'	46	104
1803	Fusion Rule 1	0.66047	0.23322	'db7'	'e'	47	103
1803	Fusion Rule 1	0.66039	0.23563	'coif5'	'e'	48	97
1803	Fusion Rule 2 WB	0.66032	0.20389	'db5'	'e'	49	136
1803	Region Based	0.65974	0.2243	'db9'	'e'	50	120
1803	Fusion Rule 2 WB	0.65961	0.2075	'db9'	'e'	51	132
1803	Region Based	0.65952	0.22851	'coif5'	'e'	52	108
1803	Fusion Rule 2 WB	0.65947	0.20737	'coif2'	'e'	53	133
1803	Region Based	0.65937	0.22628	'db7'	'e'	54	113
1803	Fusion Rule 2 WB	0.65926	0.20825	'sym5'	'e'	55	131
1803	Region Based	0.65787	0.2346	'sym5'	'e'	56	100

1803	Fusion Rule 1	0.65787	0.2346	'sym5'	'e'	57	101
1803	Fusion Rule 2 WB	0.65776	0.21051	'db7'	'e'	58	129
1803	Fusion Rule 1	0.65754	0.23346	'coif2'	'e'	59	102
1803	Region Based	0.65683	0.22652	'coif2'	'e'	60	112
1803	Region Based	0.65647	0.23519	'db5'	'e'	61	98
1803	Fusion Rule 1	0.65647	0.23519	'db5'	'e'	62	99
1803	Fusion Rule 2	0.65552	0.24737	'db2'	'u'	63	89
1803	Fusion Rule 2	0.65552	0.24737	'sym2'	'u'	64	90
1803	Fusion Rule 2 WB	0.65373	0.26368	'db2'	'r'	65	71
1803	Fusion Rule 2 WB	0.65373	0.26368	'sym2'	'r'	66	72
1803	Fusion Rule 2 WB	0.64925	0.19613	'db2'	'e'	67	141
1803	Fusion Rule 2 WB	0.64925	0.19613	'sym2'	'e'	68	142
1803	Region Based	0.64856	0.22552	'db2'	'e'	69	116
1803	Region Based	0.64856	0.22552	'sym2'	'e'	70	117
1803	Fusion Rule 1	0.64856	0.22552	'db2'	'e'	71	118
1803	Fusion Rule 1	0.64856	0.22552	'sym2'	'e'	72	119
1803	Fusion Rule 2 WB	0.64648	0.20438	'coif5'	'u'	73	135
1803	Fusion Rule 1	0.64474	0.22618	'db7'	'u'	74	114
1803	Region Based	0.64465	0.22049	'db7'	'u'	75	127
1803	Fusion Rule 2 WB	0.64439	0.19833	'db5'	'u'	76	140
1803	Fusion Rule 1	0.6443	0.22909	'coif5'	'u'	78	107
1803	Region Based	0.6443	0.22305	'coif5'	'u'	77	121
1803	Fusion Rule 1	0.64426	0.22563	'db9'	'u'	79	115
1803	Region Based	0.64392	0.21945	'db9'	'u'	80	128
1803	Fusion Rule 2 WB	0.64376	0.20239	'db9'	'u'	81	138
1803	Fusion Rule 2 WB	0.64273	0.20234	'coif2'	'u'	82	139
1803	Fusion Rule 2 WB	0.64261	0.20339	'sym5'	'u'	83	137
1803	Fusion Rule 2 WB	0.64237	0.20487	'db7'	'u'	84	134
1803	Region Based	0.64216	0.22755	'sym5'	'u'	85	109
1803	Fusion Rule 1	0.64216	0.22755	'sym5'	'u'	86	110
1803	Region Based	0.64133	0.22144	'coif2'	'u'	87	122
1803	Fusion Rule 1	0.6412	0.22735	'coif2'	'u'	88	111
1803	Region Based	0.63974	0.22924	'db5'	'u'	89	105
1803	Fusion Rule 1	0.63974	0.22924	'db5'	'u'	90	106
1803	Fusion Rule 1 L3	0.63624	0.25195	'sym5'	'r'	91	83
1803	Fusion Rule 1 L3	0.63547	0.25566	'coif5'	'r'	92	74
1803	Fusion Rule 1 L3	0.63525	0.24989	'db9'	'r'	93	85
1803	Fusion Rule 1 L3	0.63521	0.2523	'db7'	'r'	94	81
1803	Fusion Rule 1 L3	0.63377	0.2556	'db5'	'r'	95	75
1803	Fusion Rule 1 L3	0.63303	0.25908	'coif2'	'r'	96	73
1803	Fusion Rule 2 WB	0.63294	0.19158	'db2'	'u'	97	143
1803	Fusion Rule 2 WB	0.63294	0.19158	'sym2'	'u'	98	144
1803	Region Based	0.63164	0.2205	'db2'	'u'	99	123
1803	Region Based	0.63164	0.2205	'sym2'	'u'	100	124
1803	Fusion Rule 1	0.63164	0.2205	'db2'	'u'	101	125
1803	Fusion Rule 1	0.63164	0.2205	'sym2'	'u'	102	126
1803	Fusion Rule 1 L3	0.62906	0.24679	'db2'	'r'	103	92

1803	Fusion Rule 1 L3	0.62906	0.24679	'sym2'	'r'	104	93
1803	Fusion Rule 1 L3	0.61058	0.18115	'sym5'	'e'	105	152
1803	Fusion Rule 1 L3	0.60916	0.17925	'db9'	'e'	106	153
1803	Fusion Rule 1 L3	0.608	0.1823	'db7'	'e'	107	151
1803	Fusion Rule 1 L3	0.60707	0.18734	'coif2'	'e'	108	145
1803	Fusion Rule 1 L3	0.60629	0.18556	'db5'	'e'	109	147
1803	Fusion Rule 1 L3	0.6059	0.18688	'coif5'	'e'	110	146
1803	Fusion Rule 1 L3	0.60096	0.17729	'db2'	'e'	111	156
1803	Fusion Rule 1 L3	0.60096	0.17729	'sym2'	'e'	112	157
1803	Fusion Rule 1 L3	0.59548	0.17679	'sym5'	'u'	113	158
1803	Fusion Rule 1 L3	0.59329	0.17746	'db9'	'u'	114	155
1803	Fusion Rule 1 L3	0.59291	0.17836	'db7'	'u'	115	154
1803	Fusion Rule 1 L3	0.59103	0.18233	'coif5'	'u'	116	150
1803	Fusion Rule 1 L3	0.59033	0.18513	'coif2'	'u'	117	148
1803	Fusion Rule 1 L3	0.58925	0.1833	'db5'	'u'	118	149
1803	Fusion Rule 1 L3	0.58526	0.17463	'db2'	'u'	119	159
1803	Fusion Rule 1 L3	0.58526	0.17463	'sym2'	'u'	120	160
1803	Fusion Rule 1 L3	0.55168	0.50364	'coif2'	'n'	121	18
1803	Region Based	0.55166	0.53572	'db2'	'n'	122	13
1803	Region Based	0.55166	0.53572	'sym2'	'n'	123	14
1803	Fusion Rule 1	0.55166	0.53572	'db2'	'n'	124	15
1803	Fusion Rule 1	0.55166	0.53572	'sym2'	'n'	125	16
1803	Region Based	0.55092	0.54403	'sym5'	'n'	126	9
1803	Fusion Rule 1	0.55092	0.54403	'sym5'	'n'	127	10
1803	Region Based	0.55007	0.54267	'db9'	'n'	128	11
1803	Fusion Rule 1	0.55007	0.54267	'db9'	'n'	129	12
1803	Region Based	0.54993	0.54575	'coif2'	'n'	130	7
1803	Fusion Rule 1	0.54993	0.54575	'coif2'	'n'	131	8
1803	Fusion Rule 1 L3	0.54984	0.49345	'db2'	'n'	132	29
1803	Fusion Rule 1 L3	0.54984	0.49345	'sym2'	'n'	133	30
1803	Fusion Rule 1 L3	0.54965	0.50317	'coif5'	'n'	134	19
1803	Region Based	0.54921	0.54587	'db5'	'n'	135	5
1803	Fusion Rule 1	0.54921	0.54587	'db5'	'n'	136	6
1803	Fusion Rule 1 L3	0.54904	0.50139	'sym5'	'n'	137	21
1803	Region Based	0.549	0.54718	'coif5'	'n'	138	1
1803	Fusion Rule 1	0.549	0.54718	'coif5'	'n'	139	2
1803	Fusion Rule 1 L3	0.54794	0.50417	'db5'	'n'	140	17
1803	Fusion Rule 1 L3	0.54731	0.50048	'db7'	'n'	141	22
1803	Region Based	0.54645	0.54718	'db7'	'n'	142	3
1803	Fusion Rule 1	0.54645	0.54718	'db7'	'n'	143	4
1803	Fusion Rule 1 L3	0.54511	0.49901	'db9'	'n'	144	25
1803	Fusion Rule 2	0.54364	0.48286	'sym5'	'n'	145	40
1803	Fusion Rule 2	0.53628	0.4882	'db5'	'n'	146	36
1803	Fusion Rule 2	0.53515	0.49396	'coif5'	'n'	147	28
1803	Fusion Rule 2	0.53365	0.49308	'coif2'	'n'	148	31
1803	Fusion Rule 2	0.53194	0.48968	'db9'	'n'	149	33
1803	Fusion Rule 2	0.53091	0.49256	'db7'	'n'	150	32

1803	Fusion Rule 2	0.52926	0.48898	'db2'	'n'	151	34
1803	Fusion Rule 2	0.52926	0.48898	'sym2'	'n'	152	35
1803	Fusion Rule 2 WB	0.52438	0.48713	'db5'	'n'	153	37
1803	Fusion Rule 2 WB	0.51953	0.4984	'coif5'	'n'	154	26
1803	Fusion Rule 2 WB	0.51682	0.49638	'coif2'	'n'	155	27
1803	Fusion Rule 2 WB	0.51459	0.49916	'db7'	'n'	156	24
1803	Fusion Rule 2 WB	0.51345	0.50145	'sym5'	'n'	157	20
1803	Fusion Rule 2 WB	0.51253	0.48359	'db2'	'n'	158	38
1803	Fusion Rule 2 WB	0.51253	0.48359	'sym2'	'n'	159	39
1803	Fusion Rule 2 WB	0.51155	0.49992	'db9'	'n'	160	23

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

American Technology Network (ATN) Corporation. (2006, 22 October). How Night Vision Works. Retrieved October 22, 2006 from http://www.atncorp.com/HowNightVisionWorks

Brooker G. (2006). Sensors and Signals. Imaging Infrared. Retrieved November 02, 2006 from http://www.acfr.usyd.edu.au/teaching/4th-year/mech4721-Signals/material/lecture%20Imaging/01%20Infrared.pdf

Brown G.E. (October 1946). Proceedings of the National Electronics Conference: Military Application of Infrared Viewers. Chicago, Illinois.

Bullard (2006, 24 October). A Brief Military History Of Thermal Imaging. Retrieved October 24, 2006 from http://www.bullard.com/Thermalimager/Media_Info/military_history.shtml

Chaganti V. (11 April 2005). Edge Detection Of Noisy Images Using 2-D Discrete Wavelet Transform. Retrieved November 17, 2006 from http://etd.lib.fsu.edu/theses/available/etd-04192005-041139/

Chow K.C. (2004, December). Fusion of Images from Dissimilar Sensor Systems. (pp. 9 - 11). 7177. Naval Postgraduate School.

Driggers R.G., Cox P., Edwards T. (1999). Introduction to Infrared and Electro-optical Systems (pp.201). Boston: Artech House, Inc.

Farrell J. (1999). Color Imaging: Vision and Technology: Image Quality Assessment (pp. 285). John Wiley & Sons Ltd.

Force Transformation, Office of the Secretary of Defense. (2005, 5 January). The Implementation of Network Centric Warfare. Retrieved October 15, 2006 from http://www.oft.osd.mil/library/library_files/document_387_NCW_Book _LowRes. pdf.

Guyot G. (1997). Physics of the Environment and Climate (pp. 26). West Sussex, England: Wiley-Praxis Series.

Hudson R.D. (1969). Infrared Systems Engineering (pp.35-43). John Wiley & Sons, Inc. United States of America.

Imaging1. (2006, 05 November). Hand Held FLIR Vision Thermo-Sight 250D Infrared Scope Imaging Flash Camera. Retrieved November 05, 2006 from http://www.imaging1.com/thermal/IR_250.html

Infrared1. (2006, 06 November). Infrared Cameras: FLIR Thermal imaging. Retrieved November 06, 2006 from http://www.ir55.com/infrared_IR_camera.html

Jane's Avionics, Jane’s Information Group. (2005, 19 August). Electro-Optic Systems, The Tactical Night Environment. Retrieved October 25, 2006 from http://www8.janes.com.libproxy.nps.navy.mil/Search/documentView.do?docId=/content1/janesdata/yb/jav/jav_9622.htm@current&pageSelected=allJanes&keyword=image%20intensifiers&backPath=http://search.janes.com/Search&Prod_Name=JAV&

Jung C.R., Scharcanski J. (2002). Robust Watershed Segmentation using the Wavelet Transform. Proceedings of the WV Brazilian Symposium on Computer Graphics and Image Processing.

Li H., Manjunath B.S. & Mitra S.K. (1994). Multi-Sensor Image Fusion Using The Wavelet Transform. IEEE Computer Society Press. Los Alamitos, CA.

Logitech, Materials Technologies and Engineers. (2006, 23 October). Gallium Arsenide. Retrieved October 23, 2006 from http://www.logitech.uk.com/gallium_arsenide.asp.

Military.com. (2006, 05 November). NIGHT MOVES: Night Vision Technology. Retrieved November 05, 2006 from http://www.military.com/soldiertech/0,14632,Soldiertech_NightVision,,00.html

Montoro H.P. (2006, 22 October). The Technology of Night Vision. Image Intensification. Retrieved Oct 22, 2006 from http://www.photonics.com/content/handbook/2006/detectors/82233.aspx

National Aeronautics and Space Administration (NASA) Science @ Nasa, (2006, 22 October). Electromagnetic Spectrum. Retrieved October 22, 2006 from http://science.nasa.gov/newhome/help/glossfig1.hm

Nikolov S., Hill P., Bull D., Canagarajah N. (2002). Wavelets for Image Fusion. Retrieved November 14, 2006 from http://www.tallypaul.pwp.blueyonder.co.uk/papers/nikolov-wif.pdf

Norton P. (2002, September). HgCdTe Infrared Detectors. Retrieved November 03, 2006 from http://www.wat.edu.pl/review/optor/10(3)159.pdf

Northrop Grumman Electronic Systems. (2003, 7 October). Northrop Grumman Demonstrates Digital Sensor Fusion In Enhanced Night Vision Goggle At U.S. Army Field Trials. Retrieved October 03, 2006 from http://www.es.northropgrumman.com/es/eos/News.htm#.

Pedrotti F.L., Pedrotti L.S., & L. M. Pedrotti. (2007). Introduction to Optics(3rd ed) (pp. 4). New Jersey: Prentice Hall.

Polikar R. (1995, June). The Engineer’s Ultimate Guide to Wavelet Analysis: The Wavelet Tutorial. Retrieved November 14, 2006 from http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html

RCA. (1974). RCA. Electro-Optics Handbook. RCA Corporation.

Roper Scientific Inc. (2002). Introduction to Image Intensifiers for Scientific Imaging. Technical Note. Retrieved October 23, 2006 from http://www.piacton.com/pdfs/whitepapers/intensifiers.pdf.

SDC Morphological Toolbox for MATLAB. (June 2004) Retrieved November 22, 2006 from http://www.mmorph.com

Teo C.K. (2003, December). Digital Enhancement of Night Vision and Thermal Images. (pp. 9). 5355. Naval Postgraduate School.

The Mathworks. (2006, 16 November). The Wavelet Toolbox. Retrieved November 16, 2006 from http://www.mathworks.com/access/helpdesk/help/ toolbox/wavelet/wavelet.html

Therrien C.W., Scrofani J. & Krebs W.K. (October 1997). An Adaptive Technique for the Enhanced Fusion of Low-Light Visible with Uncooled Thermal Infrared Imagery. IEEE International Conference on Imaging Processing.

Wikipedia, the free encyclopedia. (2006, 21 October). Photon. Retrieved October 21, 2006 from http://en.wikipedia.org/wiki/Photon

Wikipedia, the free encyclopedia. (2006, 9 October). IMINT. Retrieved October 9, 2006 from http://en.wikipedia.org/wiki/IMINT

Wikipedia, the free encyclopedia. (2006, 22 November). Contrast. Retrieved November 22, 2006 from http://en.wikipedia.org/wiki/Contrast_(vision)

Wolin D., Johnson K. & Kipman Y. (1998). 1998 International Conference on Digital Printing Technologies: The Importance of Objective Analysis in Image Quality Evaluation. Retrieved November 07, 2006 from http://www.imagexpert.com/site-new/docs/publications/objectiveanalysis.pdf

Wong Z, Bovik A. (March 2002). A Universal Quality Index. IEEE Signal Processing Letters, Vol. 9, No. 3, March 2002

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Professor Monique P. Fargues
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

4. Professor Alfred W.M Cooper
   Department of Physics
   Naval Postgraduate School
   Monterey, California

5. Professor Yeo Tat Soon
   Temasek Defence Systems Institute
   Singapore

6. Mr Neo Tiong Tien
   The Singapore Army
   Singapore Armed Forces
   Singapore