**Chapter 1**

**INTRODUCTION**

Infrared (IR) and visible image fusion is an important technique in multi-sensor information fusion applications. Since IR sensors are able to capture thermal information in a scene that is not directly seen by human eyes, they can more clearly detect some objects in low-light, occlusion and adverse weather conditions. Visible imagery normally provides more details of the scene in the visible spectrum, and also presents more natural intensities and contrasts that are consistent with human visual perception. Integrating IR and visible information into a fused image allows us to construct a more complete and accurate description of the scene. However, due to heat emissions and differing spectral sensitivities, the relative luminance response in the IR spectrum is quite inconsistent with that in the visible spectrum, which makes the IR imagery hard to be interpreted. As a result, the fusion result of IR and visible imageries may also be visually unpleasing for human observers. Therefore, besides determining the best way to take full advantage of all information of the two source images in the fusion process, a more significant task should be to make the fused image easy to be interpreted, and thus can lead to better situation awareness.

In this work, a novel multi-scale fusion method based on a hybrid MSD transform (hybrid-MSD) to achieve better fusion results for human visual perception is presented. Unlike the previous MSD trans-forms that attempt to capture more directional information with comparatively more complex filters, the hybrid-MSD decomposes the source image into texture details and edge features at multiple scales by jointly using multi-scale Gaussian and bilateral filters. In a perceptual evaluation of different image fusion schemes, To indicate that the IR imagery serves best for target detection and recognition, whereas the visible imagery contributes most to global scene awareness. Our method manages to employ the hybrid-MSD as well as a novel asymmetrical multi-scale fusion scheme to inject the important IR spectral features into the visible image, while preserving (or properly enhancing) important perceptual cues of the background scenery and details captured from the visible spectrum. Thus, it would lead to perceptually better fusion results for human interpretation.

**Image Registration**:

Image registration is the process of overlaying two or more images of the same scene taken at different times, from different viewpoints, and/or by different sensors. It geometrically aligns two images, the reference and sensed images. The present differences between images are introduced due to different imaging conditions. Image registration is a crucial step in all image analysis tasks in which the final information is gained from the combination of various data sources.

**1.1 Objectives**

The main objective is to fuse images mainly the visible and infrared images to enhance the information of output image which could be helpful for many applications, especially for surveillance of sensitive border regions to locate enemies particularly in the forest/mountain areas Our objective is to increase the image quality by detecting and reducing the several kinds of distortion namely :

1. Impulse Noise (Salt and Pepper Noise)
2. Gaussian Noise (Amplifier Noise)
3. Poisson Noise
4. Speckle Noise

**Chapter 2**

**PROBLEM STATEMENT AND PROPOSED SOLUTION**

**2.1 Problem Statement**

In night-time environment, only limited visual information can be captured by CCD cameras under poor lightning conditions, thus making it difficult to do surveillance only by visual sensor. Therefore in this work, an image fusion technique is proposed that fuses low-light visible images and IR images capturing the same scene for better image understanding.

**2.2 Existing Approaches**

The current work begins with the survey of similar methods existing in the literature. Some of the popular methods used for image fusion are Stationary Wavelet Transform (SWT), Pulse Coupled Neural Network (PCNN) and Multi-Scale Decomposition (MSD) techniques. A detailed study of each of these techniques along with their pros and cons using experimental analysis is carried out. This section explains the theoretical part of the methods and a comparative study of the mentioned techniques are discussed in Chapter 6 – Experiments and Results. The techniques therefore explored are as follows:

**2.2.1 Stationary Wavelet Transform (SWT)**

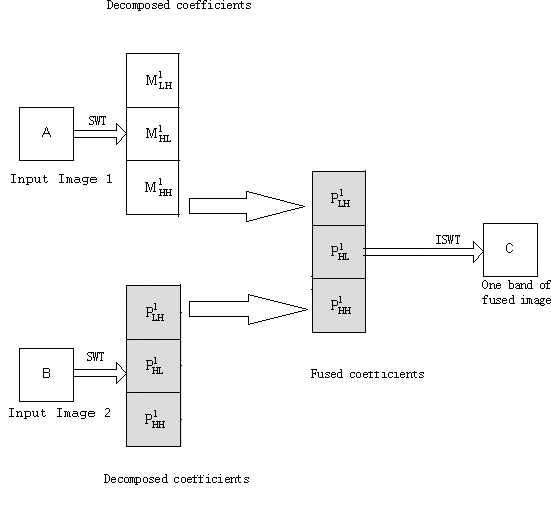
The simplest image fusion technique is to compute the average pixel-by-pixel grey level value of the source images. However, this technique leads to undesirable side effects such as contrast reduction. In the past two decades, a variety of image fusion methods were introduced such as Laplacian pyramid, Contrast pyramid, Ratio pyramid, and Discrete Wavelet Transform (DWT) and Stationary Wavelet Transform (SWT). In DWT based method, the basic idea of this method is to perform decompositions on each source image then combine all these decompositions to obtain composite representation, from which the fused image can be recovered by finding inverse transform. This method had been proved to be an effective method. However this method is not translation–invariant because of down-sampling process. If there is a movement of the object in the source images, the performance of this method will deteriorate.

However SWT is Translation invariant, without the need to carry out down-sampling process. Therefore, all approximation and detail coefficient sub-bands have the size same as the images source. In the proposed method, two fused images are firstly decomposed into four sub-bands, which are one approximation sub-band (LL) and three details sub-bands (HL, LH and HH). The recovered fused image is reconstructed by performing the Inverse Stationary Wavelet Transform.

Stationary Wavelet Transform (SWT) is similar to Discrete Wavelet Transform (DWT) but the only process of down-sampling is suppressed that means the SWT is translation-invariant.

Stationary wavelet transform is an efficient algorithm for remote sensing image fusion.

The SWT method can be described as at each level, when the high and low pass filters are applied to the data, the two new sequences have the same length as the original sequences. To do this, the original data is not decimated, however, the filters at each level are modified by padding them out with zeros. As the SWT has no dyadic decimation at each decomposition level, it yields an over complete wavelet decomposition and guarantees the result to be both aliasing-free and translation invariant. However, SWT parameters, such as the choice of the wavelet basis and number of decomposition levels, can affect its performance on image fusion. Each wavelet has its unique decomposition and reconstruction method, which leads to differences on the fusion performance. Moreover, the decomposition level also has a critical effect on the fusion performance.

****

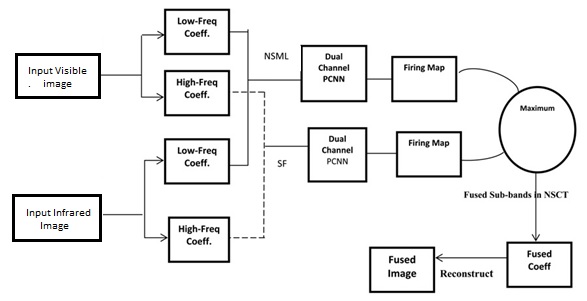
**Fig 2.1: Schematic diagram for image fusion using SWT.**

**2.2.2 Pulse Coupled Neural Network (PCNN)**

PCNN plays an important role in the image fusion process in choosing the best-quality image block for fused image.

Pulse-coupled neural network (PCNN) is a novel artificial neural network model developed by Eckhorn in 1990 and based on the experimental observations of synchronous pulse bursts in a cat’s and monkey’s visual cortex. It is characterized by the global coupling and pulse synchronization of neurons. These characteristics benefit image fusion which makes use of local image information.

To some extent, the standard PCNN structurally limits its application in image fusion. In order to make PCNN more suitable for image fusion, we improve the standard PCNN and propose the dual-channel PCNN, which can solve the problem of complication and inefficiency of PCNN methods very well.

****

**Fig 2.2: Schematic diagram for image fusion using PCNN**

Dual-Channel PCNN uses modulatory coupling rather than the more-common additive coupling. The advantage of PCNN is that a neuron with no primary input cannot be activated by the coupling input, which is an important feature in image processing. Also PCNNs are used as non-adaptive processors, thus their connectivity requirements are low, and it is practical to build them as high-speed electronic chips for high data applications such as image processing.

PCNN is a feedback network and each PCNN neuron consists of three parts: the receptive and modulation fields, information fusion pool and pulse generator. The function of receptive and modulation fields are to receive the stimulus including external inputs and surrounding neuron stimuli; information fusion pool is the place where all data are fused; the pulse generator is to generate the output pulse.

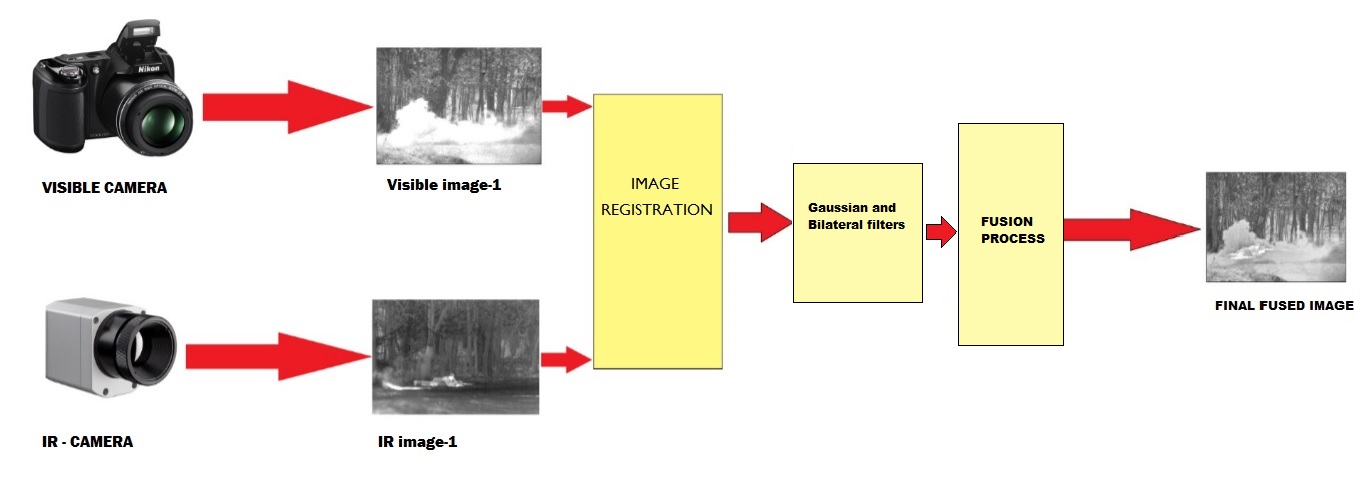
Firstly, two channels of neuron receive external stimuli and output of surrounding neurons. Furthermore, the data from these channels are weighted and mixed in the information fusion pool according to the weighting coefficients. Finally, the mixed data are released by neuron as its output with the attenuation of the threshold.

The advantage of suggested method based on pulse coupled neural network (PCNN) is that it does not require training. It uses an iterative method to obtain a proper value of linking coefficient. Its mode of information processing is much closer to the mode of human visual processing. And then PCNN has the flexible structure which can be changed according to different tasks. Additionally, the existing PCNN methods also show PCNN has the higher performance.

Experimental results show that the proposed method based on PCNN outperforms the DWT-based method.

**2.2.3 Explain the existing MSD method here**

**2.3 Proposed Approach:**

****

**Fig 2.1: Block diagram of proposed Hybrid-MSD method**

Pixel-level image fusion is conducted based on various Multi-Scale Decomposition (MSD) transforms, like Laplacian Pyramid (LAP), Gradient Pyramid (GP), Wavelet Transform and Support Vector Transform. For these MSD-based fusion methods, the MSD transforms are performed first on each source image to obtain different sub-bands containing the decomposed information of different frequencies. The corresponding sub-bands of all source images are then combined together based on certain fusion rules. Finally, the fused images can be produced by inverse MSD transforms.

The proposed Hybrid-MSD method decomposes the source images into texture details and edge features at multiple scales by jointly using multi-scale Gaussian and Bilateral filters.

**Step 1: Decomposition of Input Images**

Multi-scale decomposition is a method in which we extract the features of the image in different levels. In our proposed method we will decompose the image using Gaussian and bilateral filters.

**Step 2: Filtering using Gaussian and Bilateral**

Gaussian and Bilateral filters are two well-known filters in image processing. Gaussian filter is used to blur the image. It is widely used to reduce the image noise and to get textured detail of the image. Sigma is the standard deviation used as degree of blur.

A bilateral filter is an edge-preserving and noise-reducing filter for images. In this filter the intensity value of each pixel in an image is replaced by a weighted average of intensity values from the neighbouring pixels. This weight can be based on a Gaussian distribution. Crucially, the weights depend not only on Euclidean distance of pixels, but also on the colour intensity, depth distance, etc. This preserves sharp edges by systematically looping through each pixel and adjusting weights to the adjacent pixels accordingly. This in turn increases the visual perception of the image by preserving the edge features.

**Step 3: Image Fusion**

There are many image fusion techniques. In the proposed method the images are fused in three levels.

1. Small-scale combination

2. Large-scale combination

3. Base-level combination

**Step 3.1 Small-Scale combination:**

Small scale combination is used to integrate fine-scale features into fused image it is applied to the original image. The value of lambda is used to regulate the amount of IR image features to be injected in the visible image for better visual perception.

**Step 3.2 Large-Scale combination:**

The large-scale levels are chosen to include all the decomposed levels. At these scale levels, the decomposed large-scale edge features are fully used to identify and determine the weights of corresponding IR spectral features that would be injected into the visible image.

**Step 3.3 Base-Level combination:**

The use of base image in image fusion is that it generally provides the support information for the higher-frequency sub-bands. They are closely associated with each other in scale-spaces. So it is reasonable that the construction of the fused base image would also be related to the higher-level decomposed information in the fusion process.

**Chapter 3**

**LITERATURE SURVEY**

This section discusses the related work in the area of image fusion from various multi sensors. The study of image fusion techniques is done in both spatial and wavelet domain. The following papers provide a gist of the image fusion techniques:

1. **Title:** "Infrared and visible image fusion with the use of multi-scale edge-preserving decomposition and guided image filter”

**Author:** Wei Gan, Xiaohong Wu, Wei Wu, Xiaomin Yang, Chao Ren, Xiaohai He, Kai Liu

**Publication:** ELSEVIERInfrared Physics & Technology, Volume 72, September 2015, Pages 37–51

* In this paper a novel IR and VIS image fusion framework is proposed by combining multi-scale decomposition and guided filter. The proposed scheme could not only preserve the details of source IR and VI images but could also suppress the artefacts effectively by combining the advantages of multi-scale decomposition and guided filter.
* First, both IR and VIS images are decomposed with a multi-scale edge-preserving filter. Saliency maps of IR and VIS images are then calculated on the basis of phase congruency. Subsequently, the guided filtering is adopted to generate weighting maps. Finally, the resultant image is reconstructed with the weighting maps. Phase congruency (PC) rather than Laplace operator is adopted in this study to obtain better saliency maps, which improves the performance of the proposed method. Representative experiments show that the proposed method outperforms existing methods in image quality.

1. **Title:** "Image fusion scheme using a novel dual-channel PCNN in lifting stationary wavelet domain"

**Author:** Y. Chai, H.F. Li, J.F. Qu

**Publication:** ELSEVIER [Optics Communications](http://www.sciencedirect.com/science/journal/00304018),Volume 283, Issue 19, 1 October 2010, Pages 3591–3602

* This paper presents a new multi-source image fusion scheme based on Lifting Stationary Wavelet Transform (LSWT) and a novel dual-channel Pulse-Coupled Neural Network (PCNN). By using LSWT, we can calculate a flexible multi-scale and shift-invariant representation of registered images.
* After decomposing the original images using LSWT, a new dual-channel pulse coupled neural network, which can overcome some shortcomings of original PCNN for image fusion and putout the fusion image directly, is proposed and used for the fusion of sub-band coefficients of LSWT. In this fusion scheme, a New Sum-Modified-Laplacian(NSML) of the low frequency sub-band image, which represent the edge-feature of the low frequency sub-band image in LSWT domain, is presented and input to motivate the dual-channel PCNN.
* For the fusion of high frequency sub-band coefficients, a novel Local Neighborhood Modified-Laplacian (LNML) measurement is developed and used as external stimulus to motivate the dual-channel PCNN. This fusion scheme is verified on several sets of multi-source images, and the experiments show that the algorithms proposed in the paper can significantly improve image fusion performance, compared with the fusion algorithms such as traditional wavelet, LSWT, and LSWT–PCNN in terms of objective criteria and visual appearance.

1. **Title:** "Novel fusion method for visible light and infrared images based on NSST–SF–PCNN"

**Author:** Weiwei Kong, Longjun Zhang, Yang Lei

**Publication:** ELSEVIER Infrared Physics & Technology, Volume 65, July 2014, Pages 103–112

* This paper presented a novel fusion method for visible light and infrared images based on Non-Subsampled Shearlet Transform (NSST)–Spatial Frequency (SF)–Pulse Coupled Neural Network (PCNN).
* As a recently developed multi-resolution geometric analysis tool, NSST not only has remarked superiorities over other past conventional tools in terms of information capturing and computational costs saving, but also overcomes the lack of shift-invariance in Shearlet Transform (ST), so NSST applies to conducting the decompositions and reconstructions.
* Besides, traditional PCNN model is also upgraded to be an improved one called IPCNN in this paper to fuse the low-frequency and high-frequency sub-band coefficients. In the IPCNN structure, on the one hand, the value of the linking strength b is determined by the SF which represents the gradient features of the sub-band image; on the other hand, the time matrix is utilized to adaptively decide the iteration number of the IPCNN model, which is helpful to increase the function efficiency and save computational resources. Experimental results indicate that the proposed method performs well and has obvious superiorities over other current typical ones in both subjective visual performance and objective criteria.

1. **Title:** "Multi-Focus Image Fusion based on Stationary Wavelet Transform and extended Spatial Frequency Measurement”

**Author:**Pusit Borwonwatanadelok, Wirat Rattanapitak and Somkait Udomhunsakul

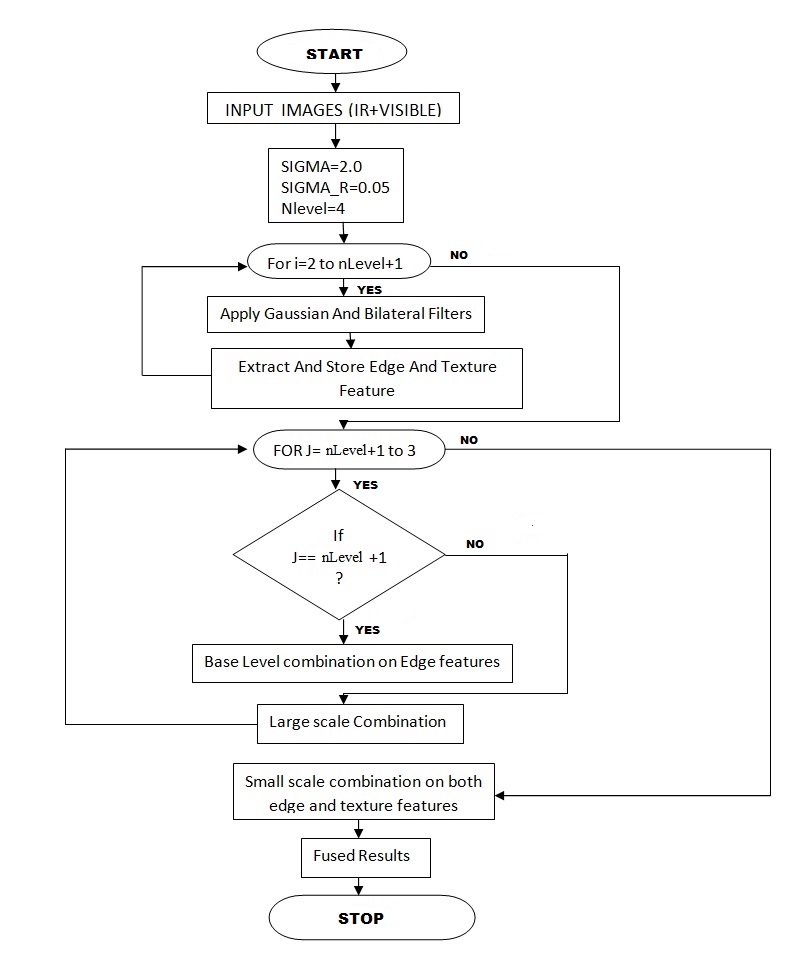
**Publication:** International Conference on Electronic Computer Technology 2009

* In this paper, we propose a multi-focus image fusion approach based on Stationary Wavelet Transform (SWT) and extended the Spatial Frequency Measurements (SFM).
* Our proposed approach, two fused images are firstly decomposed into four sub-bands, which are one approximation sub-band (LL) and three details sub-bands (HL, LH and HH).
* Next, each sub-band is partitioned into blocks and each block is identified the clearer regions by computing the focus measure using the extended Spatial Frequency Measurement (SFM).
* Finally, the recovered fused image is reconstructed by performing the Inverse Stationary Wavelet Transform. From the experimental results, we found that the proposed method outperforms the traditional Wavelet Transform and SFM based methods in terms of objective and subjective assessments.

**Chapter 4**

**ARCHITECTURE AND DESIGN**

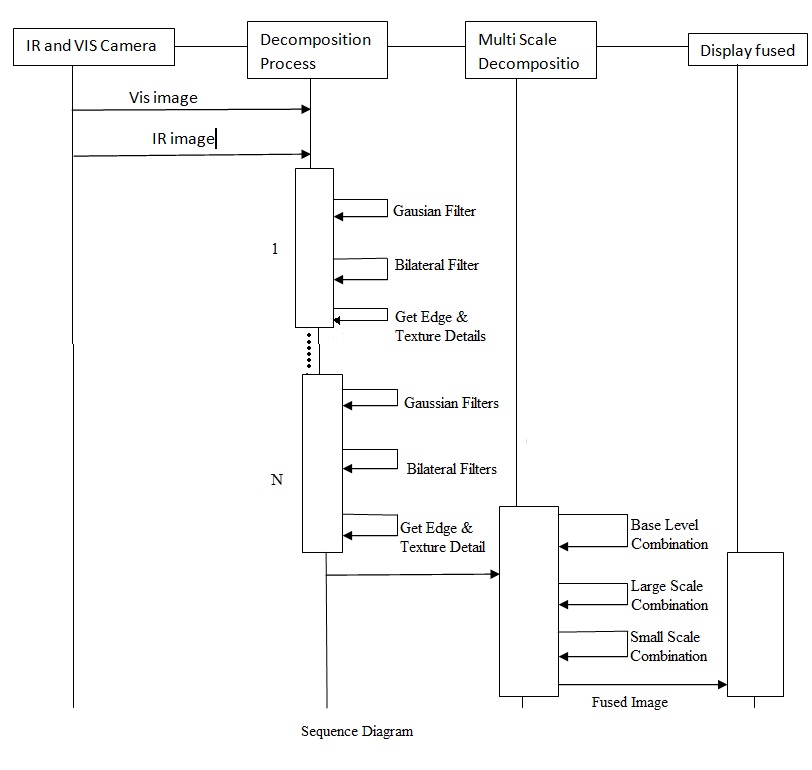
**5.2 FLOWCHART**

****

**Fig.5.2: Flowchart of the proposed Hybrid-MSD approach**



**5.3 SEQUENCE DIAGRAM**



**Fig.5.3: Sequence diagram**

**Chapter 5**

**IMPLEMENTATION**

**5.1 Software Requirements**

Software requirements for the implementation and testing

Operating System : Windows XP/07/Vista

Language : MATLAB programming language

Software Packages : MATLAB version 8.4

### 5.2 Hardware Requirements

Processor : Any Intel or AMD x86 processor

RAM : 2 GB

Input device : Keyboard, Mouse

Output device : Colour monitor

Network hardware : Network Interface Card

**5.3 Core Tools and Technologies**

This section covers the complete development matrix. It identifies the technology elements with guidelines and specifications for specific implementations. The core tools and technologies elements used in developing the project is as shown in table 4.1.

MATLAB includes development tools that help to implement the algorithm efficiently. These include the following:

* 1. **MATLAB Editor and Debugger** - Provides standard editing and debugging features, such as setting breakpoints and single stepping
  2. **Code Analyzer** - Checks the code for problems and recommends modifications to maximize performance and maintainability
  3. **MATLAB Profiler** - Records the time spent executing each line of code
  4. **Directory Reports** - Scan all the files in a directory and report on code efficiency, file differences, file dependencies, and code coverage.

|  |  |
| --- | --- |
| **Item** | **Selection** |
| Language | MATLAB Programming Language |
| Editor | MATLAB Editor |
| Debugging tools | MATLAB Debugger |

**Table 5.1: Technology Elements**

**5.4 Language Specification**

The MATLAB language supports the vector and matrix operations that are fundamental to engineering and scientific problems. It enables fast development and execution. With the MATLAB language, to develop the program and algorithms are faster than with traditional languages because this does not need to perform low-level administrative tasks, such as declaring variables, specifying data types, and allocating memory. In many cases, MATLAB eliminates the need for ‘for’ loops. As a result, one line of MATLAB code can often replace several lines of C or C++ code. At the same time, MATLAB provides all the features of a traditional programming language, including arithmetic operators, flow control, data structures, data types, Object-Oriented Programming (OOP), and debugging features.

MATLAB provides the following types of functions for performing mathematical operations and analyzing data:

1. Matrix manipulation and linear algebra
2. Polynomials and interpolation
3. Fourier analysis and filtering
4. Data analysis and statistics
5. Ordinary Differential Equations (ODEs)
6. Partial Differential Equations (PDEs)

The proposed approach comprises of two steps:

1. **MSD based on Gaussian and bilateral filters**

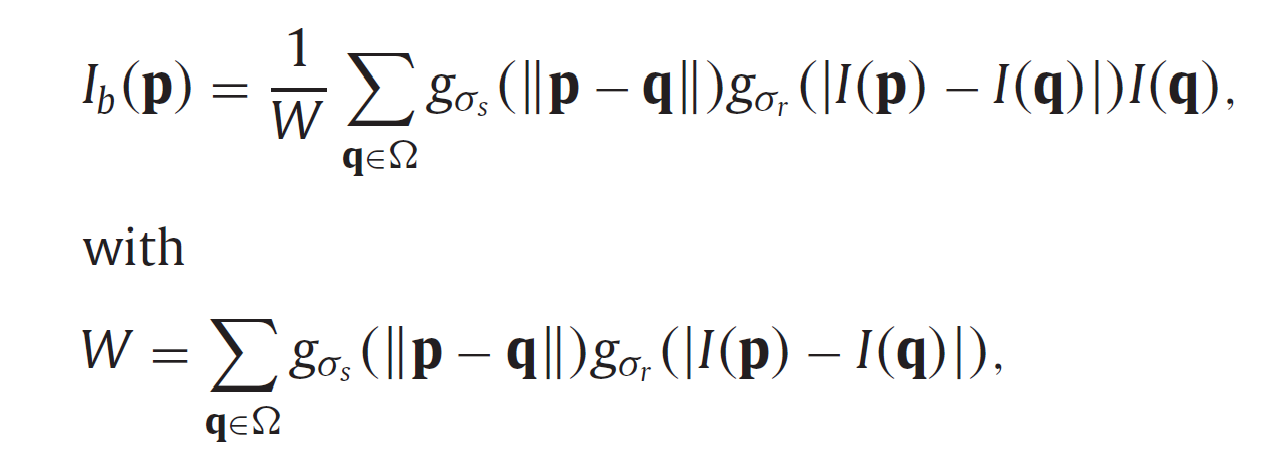
2. **Infrared and visible image fusion based on**

2.1. Small and large-scale combinations.

2.2. Base level combination.

1. **The hybrid-MSD based on Gaussian and bilateral filters**

* Gaussian and bilateral filters are both known as important filtering methods extensively used in image processing applications. The Gaussian filter is one of the basic tools for noise reduction and image smoothing.
* Gaussian smoothing is used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scales.
* A bilateral filter is a non-linear, edge-preserving and noise reducing smoothing filter for images. The intensity value at each pixel in an image is replaced by a weighted average of intensity values from nearby pixels.
* A Gaussian function denoted as *gσ(x)=exp(−x2/σ2)*
* The bilateral filtering of image ***I*** *at pixel* ***p*** is performed as***:***



Where *σ* *s* and *σ* *r* are the standard deviations of the spatial and range Gaussians, which control the influences of neighbouring pixel **q** in terms of spatial and intensity differences, respectively.

Assuming *Ig* is the corresponding Gaussian filtered image that is computed by using the spatial Gaussian *gσs,* *Ib* contains certain additional edge information compared with *Ig*. Consequently, we can obtain the fine-scale texture details by

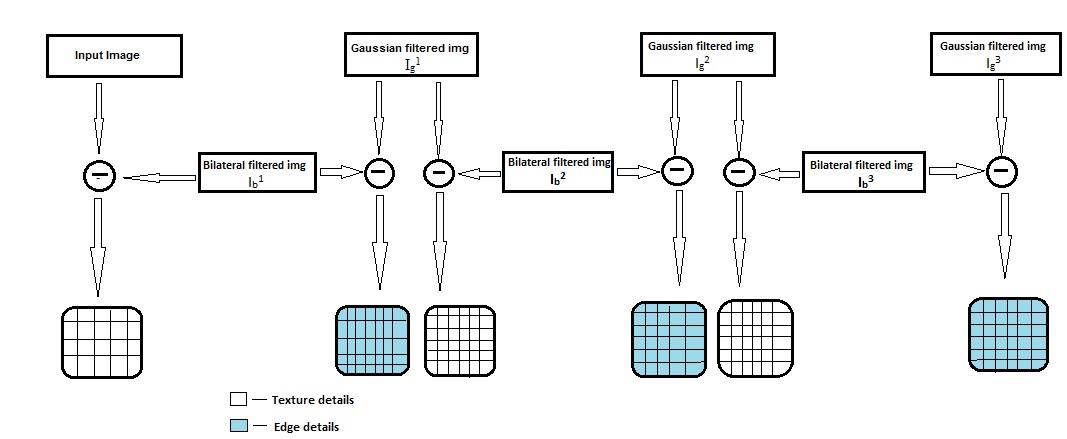
|  |
| --- |
| D0{j} **=** Ig{j-1} **−** Ib{j}, where j=2,3...N  edge features retained in *Ib* by, |

D1{j} **=** Ib{j} **−** Ig{j}.

The decomposition enables to separate image texture details from a base layer. Multi scale decomposition able extraction of image features from original images at different levels. This will increase the image visual perception and enable the method to be applied in different domains such as security like in border areas and for anti-poaching purposes. In addition to the decomposition, we also use the decomposition since it allows separating large-scale edge features from the fine-scale texture details. Furthermore, we demonstrate how hybrid-MSD can be used to fuse the infrared and visible images to allow for better visual perception.

**2. Infrared and visible image fusion based on the hybrid-MSD**

Visible images often contain more fine-scale details than the corresponding infrared images, in which the thermal information is usually captured as coarser-scale features. Therefore, there usually exists a large scale difference between the infrared and visible images, which can be measured in scale-space. Infrared images often present unnaturally high contrasts and intensities on the objects with relatively higher temperature than the surroundings. The luminance responses and contrasts of the same scene in the infrared and visible images can also be different. For these reasons, decomposing and then merging the two source image information directly by conventional MSDs without reliable feature selection may not produce a pleasant fusion result for human observation.

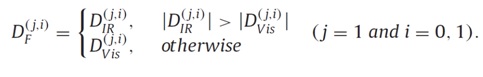


**Fig 5.1: Schematic diagram of proposed hybridisation of the MSD method using filters**

Above fig presents a three-level decomposition of the Input image in each level the input image is decomposed using Gaussian and bilateral filter. In each step we get the fine scale textured details and edge features from bilateral filter by subtraction as explained above.

**Small and large-scale combination**

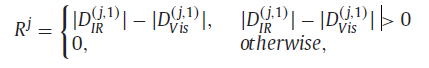
At the small-scale levels we only choose the top level (j=1) of the decomposition. Selecting absolute maximum approach is used in this scale level to integrate all the significant fine-scale features into the fused image as,



where,

DF(j,i) is resultant matrix used in further calculation

The large-scale levels are chosen to include all the decomposed levels from j=2 to j=N. At these scale levels, the decomposed large-scale edge features in D(j,1) are fully used to identify and determine the weights of corresponding IR spectral features that would be injected into the visible image. This is accomplished by computing the weighting coefficients Cj.



Rj is normalized by,



If Pj(x)>0, it indicates that the point x contains certain IR spectral information intended to be injected into the visible image at level j(2≤j≤N).

Weighting coefficients are then obtained as:

Cj = gσc \* Sλ(Pj)

The corresponding IR spectral information can be merged with the visible image information at these scale levels by

D(j,i)F = Cj D(j,i)IR + (1−Cj) D(j,i)Vis, (j=2,...,N and i=0,1).

**Base level combination**

The base image obtained by a MSD contains the coarsest scale information. The base image containing a certain amount of residual low-frequency information can well be employed to control the overall look of the fused image. Also, it requires large number of decomposition levels which would drastically increase the computational work.

Another important function of the base image in a multi-scale representation is that it generally provides the support information for the higher-frequency sub bands. They are closely associated with each other in scale-spaces. So it is reasonable that the construction of the fused base image would also be related to the higher-level decomposed information in the fusion process.

As a result, we combine the base images by taking account of the higher-level weighting coefficients as follows:

BF= Cb BIR + (1−Cb) BVis

Where the base-level weighting coefficients Cb are calculated as:

Cb = gσb \* CN

In which CN represents the weighting coefficients of level N

**Chapter 6**

**TESTING**

Verification and validation is an important phase in the development life cycle of the product. This is the phase where the errors are detected. Hence testing or verification and validation perform a very critical role for quality assurance and ensuring the reliability of the software. One definition of testing is **“The process of questioning a product in order to evaluate it”**, where the “questions” are operations the tester attempts to execute with the product, and the product answers with its behaviour in reaction to the probing of the tester.

**6.1 GOALS OF TESTING**

The foremost and basic intention behind Software Testing is to make sure that the application under test is free of bugs and errors and in case it is not, it should be identified and sent for rectification.

* Software testing plays a vital role to improve the quality of the deliverable by reporting bugs at different phases of development depending on the methodology followed for the project.
* Software testing brings with it a source of reliability that is important for any business to get hold of the market.
* Software testing ensures whether the application is ready to deliver to the client or not, so it aims to find the critical bugs early in the testing phase.
* The goal of software testing should primarily be an extension to have traceable tests with coverage of the priority requirements.
* The goal aims to ensure the flaws in application; methodologies used and are put forward in front of management in order to get it resolved.

Software testing on the whole is largely aimed at testing the application with the intent to keep the end user in mind. The goal is narrowed to the frameset where software delivery (free of bugs) within the scheduled timeline turns out to be the major element of the testing. The customer can access the application without any issue and support organizational effort should be reduced to a large extent.

**6.2 LEVELS OF TESTING**

**6.2.1 Unit Testing**

Individual components are tested to ensure that they operate correctly. Each component is tested independently. This system was tested with the set of noisy images for each module and the results were checked with the expected output. Unit testing focuses on verification effort on the smallest unit of the software design module. This is also known as MODULE TESTING. This testing is carried out during phases, each module is found to be working satisfactorily as regards to the expected output from the module.

**6.2.2 Integration Testing**

Integration testing is another aspect of testing that is generally done in order to uncover errors associated with the flow of data across interfaces. The unit-tested modules are grouped together and tested in small segment, which makes it easier to isolate and correct errors. This approach is continued until we have integrated all modules to form the system as a whole.

**6.2.3 System Testing**

System testing is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

**6.2.4 Validation Testing**

The validation testing can be defined in many ways, but a simple definition is that, validation succeeds when the software functions in a manner that can be reasonably expected by the end user.

**6.3 TYPES OF TESTING**

**6.3.1 Functional Testing**

Functional testing is a quality assurance (QA) process and a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output. Functional Testing usually describes **what** the system does.

Functional testing typically involves five steps:

• The identification of functions that the software is expected to perform.

• The creation of input data based on the specifications.

• The determination of output based on the specifications.

• The execution of the test case.

• The comparison of actual and expected outputs.

• To check whether the application works as per the customer need.

**6.3.2 Structural Testing**

Structural testing, also known as white-box testing is a method of testing the software that tests internal structures or workings of an application. In white-box testing, an internal perspective of the system, as well as programming skills are used to design test cases.

White-box testing is one of the two biggest testing methodologies used today. It primarily has three advantages:

• Side effects of having the knowledge of the source code is beneficial to thorough testing.

• Optimization of code by revealing hidden errors and being able to remove these possible defects.

• Gives the programmer introspection because developers carefully describe any new implementation.

**6.3.3 Performance Testing**

Performance testing is performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.

**6.3.4 Acceptance Testing**

Acceptance testing is a test conducted to determine if the requirements of a specification or contract are met. The acceptance test suite is run against the supplied input data or using an acceptance test script to direct the testers. Then the results obtained are compared with the expected results. If there is a correct match for every case, the test suite is said to pass. The objective is to provide confidence that the delivered system meets the requirements. The acceptance phase may also act as the final quality gateway, where any quality defects not previously detected may be uncovered.

**6.3.5 Regression Testing**

Regression testing is a type of software testing that seeks to uncover new software bugs, or **regressions**, in existing functional and non-functional areas of a system after changes such as enhancements, patches or configuration changes, have been made to them.

The intent of regression testing is to ensure that a change such as those mentioned above has not introduced new faults. One of the main reasons for regression testing is to determine whether a change in one part of the software affects other parts of the software. Common methods of regression testing include rerunning previously completed tests and checking whether program behaviour has changed and whether previously fixed faults have re-emerged.

**6.3.6 Alpha Testing**

Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

**6.3.7 Beta Testing**

Beta testing comes after alpha testing and can be considered as a form of external user acceptance testing. Versions of the software, known as beta versions, are released to groups of people so that further testing can ensure the product has few faults or bugs.

**Chapter 7**

**S**

We selected few of the standard visible and IR images which are, "meting012","Road","T1" are selected from " " to assess the performance of the proposed method. We considered other two existing methods Stationary Wavelet Tranform (SWT) and Pulse-Coupled Neural Network (PCNN) to compare the performance corresponding to the proposed hybrid MSD method.

The proposed method is compared with the existing methods by

**7.1 Subjective performance evaluation:**

The most reliable method for assessing the quality of images is through subjective testing. In subjective testing a group of people are asked to give their opinion about the quality of each image. In order to perform a subjective image quality testing, several international standards are proposed which provide reliable results. Some of these international standards are:

1. ITU-R BT.500-11
2. ITU-T P.910
3. ITU-R BT.814-1

**7.2 Objective performance evaluaton:**

The goal of objective IQA (Image Quality Assessment) is to design mathematical models that are able to predict the quality of an image accurately and also automatically. An ideal objective IQA method should be able to mimic the quality predictions of an average human observer. objective IQA methods can be classified into three categories:

1. Full-Reference Image Quality Assessment (FR-IQA) where the reference image is fully available.
2. Reduced-Reference Image Quality Assessment (RR-IQA) where only partial information about the reference image is available.
3. No-Reference Image Quality Assessment (NRIQA) where neither the reference image nor its features are available for quality evaluation.

**1. Subjective performance evaluation**

(a) (b)

(c) (d)



(e)

**Fig6.1: (a) Infrared image, (b) Visible image, (c) SWT, (d) PCNN, (e) Proposed HMSD**

1. Subjective performance evaluation

Fig6.1 shows the experimental results on "meting012".

Fig (a) and Fig (b) shows the input visible and infrared images respectively. Fig (c), Fig(d) are the fused output images using SWT and PCNN method respectively. Fig (e) is the fused output from proposed Hybrid-MSD method. The man behind the smoke can't be seen in visible image but the smoke and surroundings are clear. In infrared image the man behind the smoke can be seen clearly but the smoke can't be seen and the surrounding has less texture detail compared to the visible image. The smoke, the man hiding behind the smoke and the surrounings are clear in the fused image. All the fusion methods extract the input image features and display it, but there will be difference in contrast, details and visual quality of the fused images.

Fig (c) shows the fused output based on SWT method, in which we can notice that the smoke and the surrounding texture details are not clear and the background in the images does not have a clear picture of tree leafs and the edges of the trees are not as clear as in the MSD method although the image quality is better than the PCNN fused image.

Fig (d) shows the fused output based on PCNN method, in which the trees are dark, the edges of the tree are not clear and the image quality is not good. Smoke in the fused image can't be seen properly, it has less features of the input images when compared to the results of the proposed method and is not ideal for visual perception.

Fig (e) shows the fused output based on the proposed Hybrid-MSD method, the resultant image has more texture details compared to other methods and we can clearly see the smoke, man hiding in smoke and the trees. It has high visual perception compared to other methods.

1. (b)

(c) (d)

****

(e)

**Fig6.2: (a) Infrared image, (b) Visible image, (c) SWT, (d) PCNN, (e) Proposed HMSD**

Above Fig6.2 shows experiment result of "Road" image. Car and traffic lights are clear in the IR image whereas the advertising board is clear but the car and traffic lights are not clear in the visible image.

Fig (c) shows the fused image based on SWT method, car, traffic light and advertising board can be seen in the fused image but there is some degree of blur and the edges of the advertising board are not preserved. It has lesser image quality compared to Fig (e), although it has better image quality than fused image based on PCNN.

Fig (d) shows the fused image based on PCNN method, in the resultant fused image the car, traffic lights and advertising board are not clear and it has too poor image quality and lighting.

Fig (e) shows the fused image based on Hybrid-MSD method, resultant image has clear picture of car, traffic lights and advertising board. It has better visual perception compared to other two methods.

1. (b)

 ****

(c) (d)

**Fig6.3: (c-e) shows experimental result for image "T1"**.

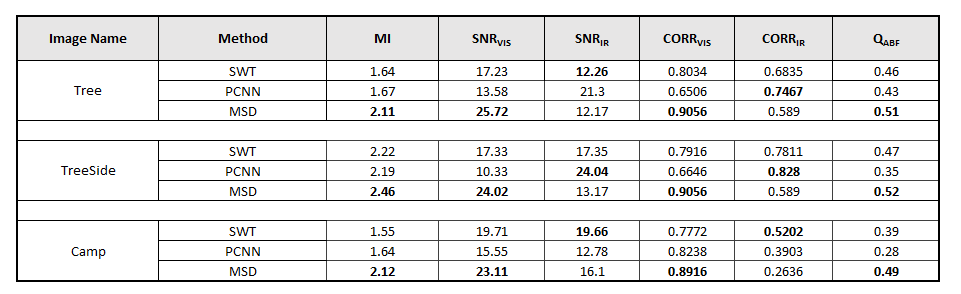
As like the above the proposed method has better visual perception then the other two methods. The light sources and the car are clear in the fused image based on Hybrid-MSD method. There is a clear difference in edges.

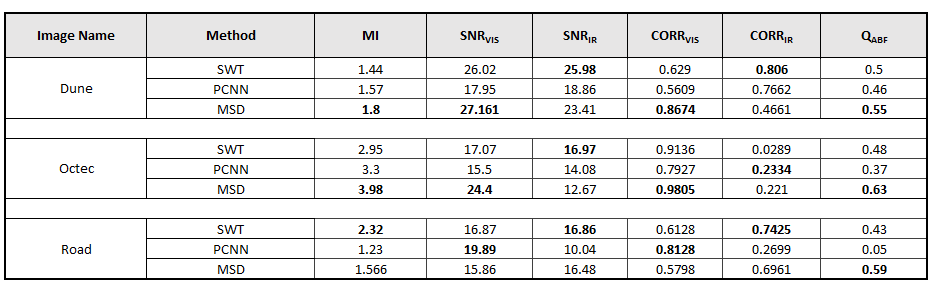
****

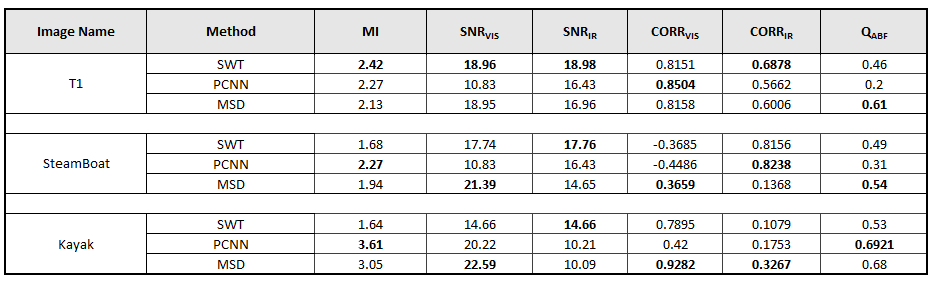
(e)

**Fig6.3: (a) Infrared image, (b) Visible image, (c) SWT, (d) PCNN, (e) Proposed HMSD**

2. Objective performance evaluation

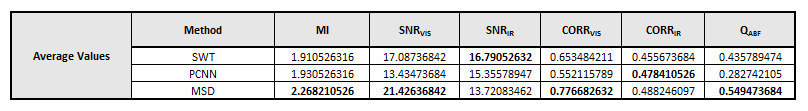






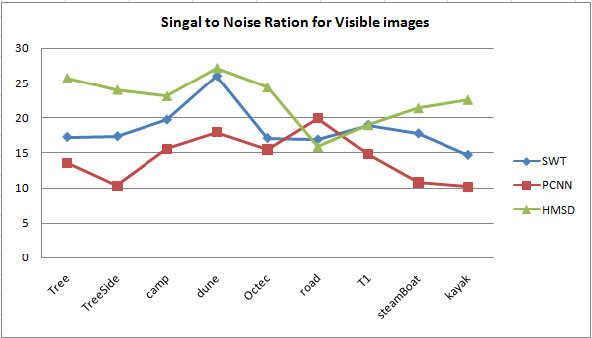
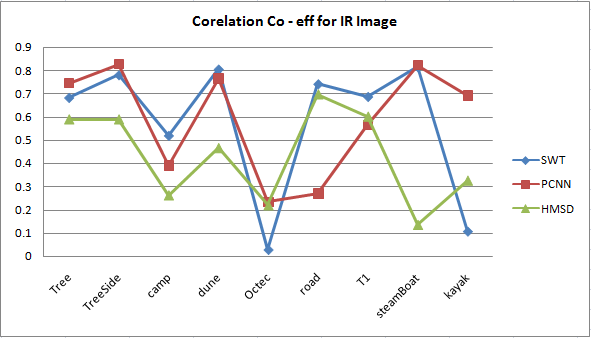
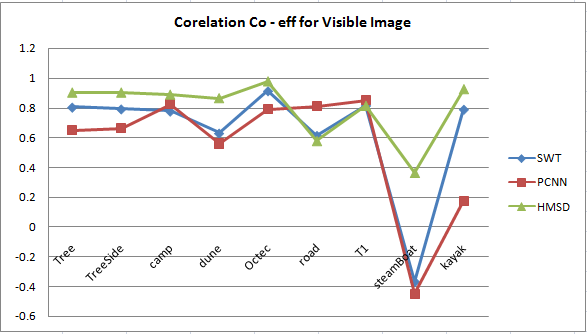
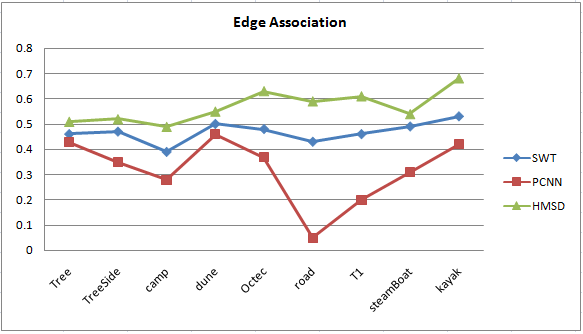
The Objective Performance Evaluation characteristics are shown in the above tables. The tables show the values of various performance evaluation factors that can be used to evaluate and determine the best method for image fusion. The tables shows the performance matrix of various images for the previously mentioned methods (i.e, SWT, PCNN, Hybrid MSD). The performance evaluation factors are: Mutual Information (MI), Signal to Noise Ratio (SNR), Correlation Coefficient (CORR) and Edge Association (QAB/F).

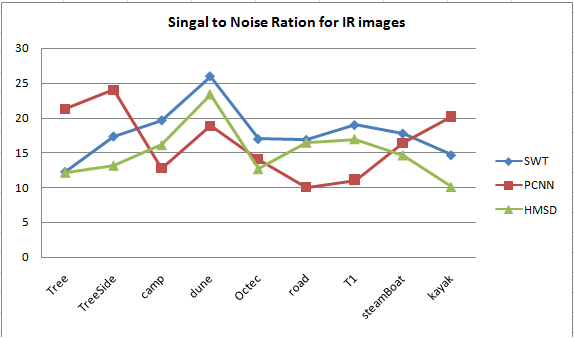
Mutual informationis adopted in the information theory-based metric. The metric with mutual information measures the amount of source image information preserved in the fused image after fusion processing. The QAB/F metric is the feature-based metric,which measures how well the amount of edge information is kept in the fused images. Higher values of these three metrics indicate better fusion quality of the fused image. Signal-to-noise ratio (abbreviated SNR) is a measure that compares the level of a desired [signal](https://en.wikipedia.org/wiki/Signal_(electrical_engineering)" \o "Signal (electrical engineering)) to the level of background [noise](https://en.wikipedia.org/wiki/Noise_(signal_processing)" \o "Noise (signal processing)). It is defined as the ratio of signal power to the noise power. A ratio higher than 1:1 indicates more signal than noise. The correlation coefficient (abbreviated CORR) is a measure that determines the degree to which two variables' movements are associated. The range of values for the correlation coefficient is -1.0 to 1.0. A correlation of -1.0 indicates a perfect [negative correlation](http://www.investopedia.com/terms/n/negative-correlation.asp), while a correlation of 1.0 indicates a perfect [positive correlation](http://www.investopedia.com/terms/p/positive-correlation.asp).

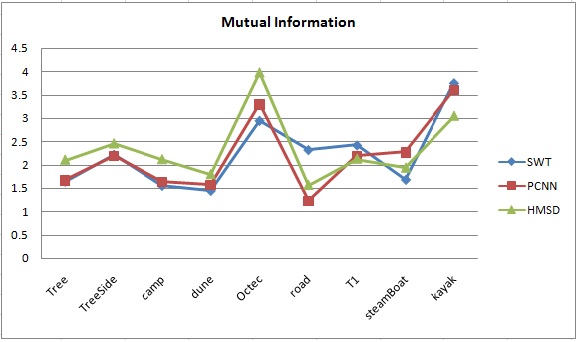
From the above table it can be seen that the average performance matrix values of the proposed method outperform the others in most of the evaluation factors. We can also see thatthe average contributions of VIS image and IR image with the proposed method are larger than the average contributions with othermethods in terms of QAB/F and MI . Moreover, the average contributions of IR image with the proposed method are the best among the SNR and CORR.

These indicate that generally the performance of the proposed method is the best among the competing methods. The mutual information, edge characteristics, and structural information of the source images are well-transferred into the fused image. This leads to the conclusion that the proposed method has the best performance among the compared methods on the whole.

**Graphs**







**Chapter 8**

**CONCLUSION**

In this project, a novel method to fuse Infrared (IR) and visible images has been proposed based on combining rule for a multi scale decomposition based image fusion. We use statistical filter data of a visible image for transformation to refine major information of IR image.

The filters can be used to identify and select important IR spectral features from the infrared image to inject them into the visible image. By further employing different combination algorithms adaptively according to different information scale levels in the fusion process, we can preserve or properly enhance the background scenery and details from the visible image which provide important perceptual cues for human observation.

Experimental results demonstrate that the proposed fusion method is able to provide perceptually better fusion results compared with various other pixel-based multi-scale fusion algorithms.

**APPENDIX**

**Source Code:**

**Pulse-Coupled Neural Network (PCNN) Source Code**

BETA=0.5; % note BETA1=BETA2; SO WE USE BETA FOR CONVENIENCE

M=[0.707,1,0.707;1,0,1;0.707,1,0.707];

DELTA=1;

ALPHAT=0.012;

VT=4000;

img1WT=conv2(img1,M,'same');

img2WT=conv2(img2,M,'same');

SUMPIXEL=ROW\*COL; %Total number of pixels

JUDFIR=zeros(ROW,COL); % JUDFIR IS USED TO JUDGE WHETHER THE NEURON(I,J) IS FIRED

THR=500\*ones(ROW,COL);

FI=zeros(ROW,COL);

for(itime=1:10)

for(i=1:ROW)

for(j=1:COL)

if(JUDFIR(i,j)~=1) % THE NEURON SHOULD NOT BE FIRED YET.

HA=img1WT(i,j)+img1(i,j);

HB=img2WT(i,j)+img2(i,j);

FI(i,j)=(1+BETA\*HA)\*(1+BETA\*HB)+DELTA;

if (FI(i,j)>THR(i,j))

JUDFIR(i,j)=1;

else

THR(i,j)=THR(i,j)\*exp(-ALPHAT);

end

end %%% end of JUDFIR

end % end j

end % end i

if(sum(sum(JUDFIR))==SUMPIXEL)

fprintf(1,'fused successfully');

break

end

end % end itime

FI=FI/max(max(FI))\*255;

figure;

imshow(uint8(FI));

**Stationary Wavelet Transform (SWT) Source Code**

%function[] = SWTFIuse1L\_demo()

% Image fusion by (one level)discrete stationary wavelet transform

close all;

clear all;

home;

% insert images

img1 = double(imread('Steamboat\_Vis.jpg'));

img2 = double(imread('Steamboat\_IR.jpg'));

figure(1);

subplot(121);

\imshow(img1,[]);

subplot(122);

imshow(img2,[]);

img1=imresize(img1,[510 506]);

img2=imresize(img2,[510 506]);

% image decomposition using discrete stationary wavelet transform

[A1L1,H1L1,V1L1,D1L1] = swt2(img1,1,'sym2');

[A2L1,H2L1,V2L1,D2L1] = swt2(img2,1,'sym2');

% fusion start

AfL1 = 0.5\*(A1L1+A2L1);

D = (abs(H1L1)-abs(H2L1))>=0;

HfL1 = D.\*H1L1 + (~D).\*H2L1;

D = (abs(V1L1)-abs(V2L1))>=0;

VfL1 = D.\*V1L1 + (~D).\*V2L1;

D = (abs(D1L1)-abs(D2L1))>=0;

DfL1 = D.\*D1L1 + (~D).\*D2L1;

% fused image

FI = iswt2(AfL1,HfL1,VfL1,DfL1,'sym2');

figure(2);

imshow(FI,[]);

**Source Code for Proposed Hybrid-MSD Method**

clc;

nLevel = 4; %number of decompositions

lambda=30;

img1=imread('meting1\_Vis.bmp');%read Vis image

img2=imread('meting1\_IR.bmp');%read IR image

%display input images

figure(1),title('Visible image'),imshow(img1);

figure(2),title('Infrared Image'),imshow(img2);

img1=double(img1);

img2=double(img2);

%% ---------- Hybrid Multi-scale Decomposition --------------

sigma = 2.0; %std deviation of spartial gaussian

sigma\_r = 0.05; %std deviation of range gaussian

k = 2;

%Gaussian and bilateral filter applied to Visible image

M1 = cell(1, nLevel+1);

M1L = cell(1, nLevel+1);

M1{1} = img1;

M1L{1} = img1;

M1D = cell(1, nLevel+1);%decomposed textured detail stored

M1E = cell(1, nLevel+1);%decomposed edge detail stored

sigma0 = sigma;

for j = 2:nLevel+1,

w = floor(3\*sigma0);

%apply gaussian filter

h = fspecial('gaussian', [2\*w+1, 2\*w+1], sigma0);

M1{j} = imfilter(M1{j-1}, h, 'symmetric');

%apply bilateral filter

M1L{j} = 255\*bfilter2(M1L{j-1}/255,w,[sigma0, sigma\_r/(k^(j-2))]);

%get the textured detail from substraction of prev gaussian with bilateral filter

M1D{j} = M1{j-1} - M1L{j};

%get the edge features

M1E{j} = M1L{j} - M1{j};

sigma0 = k\*sigma0;

end

%Gaussian and bilateral filter applied to Visible image

M2 = cell(1, nLevel+1);

M2L = cell(1, nLevel+1);

M2{1} = img2;

M2L{1} = img2;

M2D = cell(1, nLevel+1);%decomposed textured detail stored

M2E = cell(1, nLevel+1);%decomposed edge detail stored

sigma0 = sigma;

for j = 2:nLevel+1,

w = floor(3\*sigma0);

h = fspecial('gaussian', [2\*w+1, 2\*w+1], sigma0);

M2{j} = imfilter(M2{j-1}, h, 'symmetric');

M2L{j} = 255\*bfilter2(M2L{j-1}/255,w,[sigma0, sigma\_r/(k^(j-2))]);

%get the textured detail from substraction of prev gaussian with bilateral filter

M2D{j} = M2{j-1} - M2L{j};

%get the edge features

M2E{j} = M2L{j} - M2{j};

sigma0 = k\*sigma0;

end

%%Multi-scale Combination

for j = nLevel+1 : -1 : 3

b2 = abs(M2E{j});

b1 = abs(M1E{j});

R\_j = max(b2-b1, 0);

Emax = max(R\_j(:));

P\_j = R\_j/Emax;

C\_j = atan(lambda\*P\_j)/atan(lambda);

% Base level combination

sigma0 = 2\*sigma0;

if j == nLevel+1

w = floor(3\*sigma0);

h = fspecial('gaussian', [2\*w+1, 2\*w+1], sigma0);

lambda\_Base = lambda;

%lambda\_Base = 30;

C\_N = atan(lambda\_Base\*P\_j)/atan(lambda\_Base);

C\_N = imfilter(C\_N, h, 'symmetric');

MF = C\_N.\*M2{j} + (1-C\_N).\*M1{j};

end

% Large-scale combination

sigma0 = 1.0;

w = floor(3\*sigma0);

h = fspecial('gaussian', [2\*w+1, 2\*w+1], sigma0);

C\_j = imfilter(C\_j, h, 'symmetric');

D\_F = C\_j.\*M2E{j}+ (1-C\_j).\*M1E{j};

MF = MF + D\_F;

D\_F = C\_j.\*M2D{j}+ (1-C\_j).\*M1D{j};

MF = MF + D\_F;

end

% Small-scale combination

sigma0 = 0.5;

w = floor(3\*sigma0);

h = fspecial('gaussian', [2\*w+1, 2\*w+1], sigma0);

C\_0 = double(abs(M1E{2}) < abs(M2E{2}));

C\_0 = imfilter(C\_0, h, 'symmetric');

D\_F = C\_0.\*M2E{2}+ (1-C\_0).\*M1E{2};

MF = MF + D\_F;

C\_0 = abs(M1D{2}) < abs(M2D{2});

% C\_0 = imfilter(C\_0, h, 'symmetric');

D\_F = C\_0.\*M2D{2}+ (1-C\_0).\*M1D{2};

MF = MF + D\_F;

% Fusion Result

FI = ImRegular(MF); % The intensities are regulated into [0, 255]

FI = max(min(FI,255), 0);

%paraShow.fig = 'Fusion result';

%ShowImageGrad(MF, paraShow);

FI=uint8(FI);

figure(3),title('Fused image'),imshow(FI);

**Source Code for Bilateral Filter**

function B = bfilter2(A,w,sigma)

% Apply either grayscale or color bilateral filtering.

if size(A,3) == 1

B = bfltGray(A,w,sigma(1),sigma(2));

else

B = bfltColor(A,w,sigma(1),sigma(2));

end

% Implements bilateral filtering for grayscale images.

function B = bfltGray(A,w,sigma\_d,sigma\_r)

% Pre-compute Gaussian distance weights.

[X,Y] = meshgrid(-w:w,-w:w);

G = exp(-(X.^2+Y.^2)/(2\*sigma\_d^2));

% Apply bilateral filter.

dim = size(A);

B = zeros(dim);

for i = 1:dim(1)

for j = 1:dim(2)

% Extract local region.

iMin = max(i-w,1);

iMax = min(i+w,dim(1));

jMin = max(j-w,1);

jMax = min(j+w,dim(2));

I = A(iMin:iMax,jMin:jMax);

% Compute Gaussian intensity weights.

H = exp(-(I-A(i,j)).^2/(2\*sigma\_r^2));

% Calculate bilateral filter response.

F = H.\*G((iMin:iMax)-i+w+1,(jMin:jMax)-j+w+1);

B(i,j) = sum(F(:).\*I(:))/sum(F(:));

end

end

% Implements bilateral filter for color images.

function B = bfltColor(A,w,sigma\_d,sigma\_r)

% Convert input sRGB image to CIELab color space.

if exist('applycform','file')

A = applycform(A,makecform('srgb2lab'));

else

A = colorspace('Lab<-RGB',A);

end

% Pre-compute Gaussian domain weights.

[X,Y] = meshgrid(-w:w,-w:w);

G = exp(-(X.^2+Y.^2)/(2\*sigma\_d^2));

% Rescale range variance (using maximum luminance).

sigma\_r = 100\*sigma\_r;

% Apply bilateral filter.

dim = size(A);

B = zeros(dim);

for i = 1:dim(1)

for j = 1:dim(2)

% Extract local region.

iMin = max(i-w,1);

iMax = min(i+w,dim(1));

jMin = max(j-w,1);

jMax = min(j+w,dim(2));

I = A(iMin:iMax,jMin:jMax,:);

% Compute Gaussian range weights.

dL = I(:,:,1)-A(i,j,1);

da = I(:,:,2)-A(i,j,2);

db = I(:,:,3)-A(i,j,3);

H = exp(-(dL.^2+da.^2+db.^2)/(2\*sigma\_r^2));

% Calculate bilateral filter response.

F = H.\*G((iMin:iMax)-i+w+1,(jMin:jMax)-j+w+1);

norm\_F = sum(F(:));

B(i,j,1) = sum(sum(F.\*I(:,:,1)))/norm\_F;

B(i,j,2) = sum(sum(F.\*I(:,:,2)))/norm\_F;

B(i,j,3) = sum(sum(F.\*I(:,:,3)))/norm\_F;

end

end

% Convert filtered image back to sRGB color space.

if exist('applycform','file')

B = applycform(B,makecform('lab2srgb'));

else

B = colorspace('RGB<-Lab',B);

end

**Source Code For Performance Evaluation**

Mutual Information

Function mutural\_informationR = mutural\_info(matrixA,matrixB,matrixF, grey\_level)

HA = entropy\_fusion(matrixA,grey\_level);

HB = entropy\_fusion(matrixB,grey\_level);

HF = entropy\_fusion(matrixF,grey\_level);

HFA = Hab(matrixF,matrixA,grey\_level);

HFB = Hab(matrixF,matrixB,grey\_level);

MIFA=HA+HF-HFA;

MIFB=HB+HF-HFB;

mutural\_informationR=MIFA+MIFB;

%Source code for entropy\_fusion Function

function entropyR=entropy\_fusion(grey\_matrix,grey\_level)

[row,column,r]=size(grey\_matrix);

total=row\*column;

counter=zeros(1,grey\_level);

grey\_matrix=grey\_matrix+1;

for i=1:row

for j=1:column

indexx= grey\_matrix(i,j);

if(uint8(indexx)~=0)

counter(uint8(indexx))=counter(uint8(indexx))+1;

end

end

end

total= sum(counter(:));

index = find(counter~=0);

p = counter/total;

entropyR= sum(sum(-p(index).\*log2(p(index))));

%Source code for Hab Function

function HabR=Hab(grey\_matrixA,grey\_matrixB,grey\_level)

[row,column]=size(grey\_matrixB);

counter = zeros(256,256);

grey\_matrixA=grey\_matrixA+1;

grey\_matrixB=grey\_matrixB+1;

for i=1:row

for j=1:column

indexx = uint8(grey\_matrixA(i,j));

indexy = uint8(grey\_matrixB(i,j));

if(indexx~=0 && indexy~=0)

counter(indexx,indexy) = counter(indexx,indexy)+1;%ÁªºÏÖ±·½Í¼

end

end

end

total= sum(counter(:));

index = find(counter~=0);

p = counter/total;

HabR = sum(sum(-p(index).\*log2(p(index))));