DEVELOPMENT OF CANNY OPERATOR BASED GUI FOR THE MEASUREMENT OF OPTICAL RETURN LOSS

By

EKUNDAYO OLUWASEUN ADETIBA

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
FACULTY OF ENGINEERING
AHMADU BELLO UNIVERSITY, ZARIA
NIGERIA

MARCH, 2016

DEVELOPMENT OF CANNY OPERATOR BASED GUI FOR THE MEASUREMENT OF OPTICAL RETURN LOSS

By

Ekundayo Oluwaseun ADETIBA

B.Eng. (FUTA) 2008

P14EGCM8002

dayohigh@gmail.com

A DISSERTATION SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES, AHMADU BELLO UNIVERSITY, ZARIA IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF A MASTER OF SCIENCE (MSc) DEGREE IN COMMUNICATION ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

FACULTY OF ENGINEERING

AHMADU BELLO UNIVERSITY, ZARIA

NIGERIA

MARCH, 2016

DECLARATION

I, ADETIBA, Oluwaseun Ekundayo, hereby declare that this dissertation titled **"Development of Canny Operator Based GUI for the Measurement of Optical Return Loss"** was carried out by me under the supervision of Prof. B. G. Bajoga and Dr. S. M. Sani as part of the requirements for the award of degree of Master of Science in Communication Engineering, A. B.U-Zaria. To the best of my knowledge, this research work has never been submitted anywhere for the award of any certificate. All literatures used and cited in this report, are duly acknowledged by means of references.

Ekundayo Oluwaseun ADETIBA	______________________	__________________
	Signature	Date

CERTIFICATION

This dissertation titled "Development of Canny Operator Based GUI for the Measurement of Optical Return Loss" meets the requirements for the award of degree of Master of Science (MSc) in Communication Engineering by Ahmadu Bello University, Zaria and it is approved for its contribution to knowledge and literature.

Chairman, Supervisory Committee (Professor B. G. Bajoga)	…………………… Signature	**……………………..** Date
Member, Supervisory Committee (Dr. S. M. Sani)	…………………… Signature	**……………………..** Date
Head of Department (Dr. Y. Jibril)	…………………… Signature	**……………………..** Date
Dean Postgraduate School (Professor K. Bala)	…………………… Signature	**……………………..** Date

DEDICATION

This dissertation is dedicated to God Almighty.

ACKNOWLEDGEMENT

I acknowledge God almighty for His mercies and loving kindness towards me which are beyond measure for which I remain eternally grateful.

I also extend my gratitude to my parents Mr. and Mrs. Adetiba for their love, support and prayers, my brother Dr. T. E. Adetiba for being an inspiration and my loving wife Tosin whose encouragement spurred me on every inch of the way, May God almighty preserve them all.

Furthermore, I express my sincere gratitude to the chairman of my supervisory team Professor B.G Bajoga who brought his numerous years of professional and academic experience to bear on the outcome of this work, I also thank Dr. S. M. Sani, a member of my supervisory team who literarily walked me from first principles through the processes of scholastic erudition which have in no small way guided me every step of the way, May God almighty bless them both richly.

My sincere gratitude also goes to the Head of Department Dr. Y. Jibril for fostering a favourable academic climate for learning and the deployment of his administrative acumen in mitigating the various encumbrances encountered, may God almighty bless him richly.

Also, my sincere gratitude to Dr. Tekanyi who has been there to attend to any and every concerns related to this work with keen interest and empathy at any time. May God almighty reward him abundantly.

My appreciation, in no particular order of gratitude, also goes to Prof. M.B Mu'azu, Prof. B. Jimoh, Dr. S. Garba, Dr. A. D. Usman, Dr. Abu-Bilal, Dr. T. Sikiru, Dr. Umoh for their immense contributions. I am also grateful to all the academic staff of the Department; Engr. Olarinoye, Engr. Zanna, Engr. Josiah, Engr. Taiwo, Engr. E. Okafor, Engr. Abdullahi, Engr. Adamu, Engr. B. O. Sadiq, Engr. Mu'azu, Engr. A. I. Abdullahi, Engr. S. Musa, Engr. P. Okorie, Engr. A. M.

Kabir. My gratitude also goes to all the administrative and support staff for their cooperation and support.

My sincere gratitude also goes to all my colleagues and friends Fred, Francis, Maryam, Otori Abdulaziz, Habila and others not mentioned herein for their moral support and encouragement.

ADETIBA Oluwaseun Ekundayo.
March, 2016

TABLE OF CONTENTS

Development of Canny Operator Based GUI for the Measurement of Optical Return Loss
Declaration i
Certification ii
Dedication iii
Acknowledgement iv
Table of Contents vi
List of Figures..ix
List of Tables..x
List of Abbreviations..xi
Abstract...xii

1.0 CHAPTER ONE: INTRODUCTION
1.1 Background. 1
1.1.1 Introduction to Optical Fiber Communications System. 2
1.1.2 Digital Image Operations. 3
1.2 Aim and Objectives 5
1.3 Statement of Problem 5
1.4 Methodology 6
1.5 Dissertation Outline 8

CHAPTER TWO: LITERATURE REVIEW
2.1 Introduction 9
2.2 Review of Fundamental Concepts 9
2.2.1 Wave Propagation. 9
2.2.2 Optical Fiber profiles and modes. 10
2.2.3 Optical fiber attenuation 11
2.2.4 Optical Return Loss (ORL) 12
2.2.5 Edge Detection 19
2.2.6 Image Convolution theorem 22
2.2.7 Image Histogram 28
2.2.8 Image Segmentation 29
2.2.9 Image Similarity Assessment 33
2.3 Review of Similar Works 41

3.0 CHAPTER THREE: MATERIALS AND METHODS
3.1 Introduction 46
3.2 Image Histogram...46
3.3 Edge Detection Operators..47
3.4 Pixel-Based Segmentation. 49
3.5 Development of Connector Endface Return loss model equation. 49

4.0 CHAPTER FOUR: RESULTS AND DISCUSSION
4.1 Introduction......51
4.2 Edge Detection Methods Evaluation......51
4.3 Fiber optic Connector ORL Model......53
4.4 Connector ORL Measurement GUI.......56
4.5 Validation of Connector ORL Model......57

5.0 CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS
5.1 Conclusion......60
5.2 Significant Contributions......60
5.3 Limitation......61
5.4 Recommendation for further work......61

REFERENCES......62
APPENDICES......68
Appendix A......68
Source Code for the Implementation Of Edge Detection Operators......68
Appendix B......70
Matlab Source Code for the Plot of Detected Edges per Section of Connector EndfacProfile......70
Appendix C......71
Screenshot of the data statistics tool of plot of edge detections per section of the connector endface profile.......71
Appendix D......72
Laplacian of Gaussian Filter Code......72
Appendix E......73
Image Profile Sections of Canny Operator Output......73
Appendix F......76
Reference MQJ and IUT connector profile images and their OTDR measured Connectors Return Loss......76
Appendix G......78
MalabM-file source code for connector return loss equation parameters......78
Appendix H......79
Matlab Graphic User Interface [GUI] Source Code......80
Appendix I......85
Singlemode Optical Return Loss Measurement GUI Interface.......85
Appendix J......86
Multimode Optical Return Loss Measurement GUI Interface.......86
Appendix K......87
Setup of OTDR and Inspection microscope measurements.......87
Appendix L......88
Typical Optical Fiber Cable Manufacturer’s Specifications......88

List of Figures

Figure 1.1:	Generic Communication Network Model	1
Figure 1.2:	A typical Fiber optic link	2
Figure 1.3:	Illustration of Point Operation.	4
Figure 1.4:	Illustration of Local Operation	4
Figure 1.5:	Illustration of Global Operation	4
Figure 2.1:	Classification of Optical guided modes	10
Figure 2.2:	A Simple OCWR Measurement	14
Figure 2.3:	OTDR Trace of Power Loss against distance from the Launch Point	16
Figure 2.4:	Event and Attenuation Dead zone	16
Figure 2.5:	Setup for ORD Measurements	17
Figure 2.6:	Sample ORD measurement Plot	18
Figure 2.7:	Intensity Profile and First Derivative results.	20
Figure 2.8:	Intensity Profile and, First and Second Derivative Results	22
Figure 2.9:	Image Histograms	28
Figure 2.10	Illustration of Pixel Based Segmentation	31
Figure 2.11:	Illustration of Feature Based Similarity Measure	34
Figure 3.1:	Flow chart of the image histogram equalization process.	46
Figure 3.2:	Grayscale Singlemode connector Endface and Histograms.	47
Figure 3.3:	Flowchart of edge detection implementation	48
Figure 3.4:	Flowchart of Pixel Based Segmentation.	49
Figure 3.5:	Flowchart of Image Similarity Coefficient Extraction.	50
Figure 4.1:	Graphical Outputs of Edge Detection Operators	51
Figure 4.2:	Plot of the sum of detected edges along each section of the connector Endface.	54
Figure 4.3:	Screen Grab of the Matlab Base ORL computation	55
Figure 4.4:	Snippet of Developed Matlab GUI Backend Symbolic Math Implementation	56
Figure 4.5:	Developed Connector ORL Measurement GUI.	57
Figure 4.6:	Schematic of Connector endface Capture setup.	58

List of Tables

Table 4.1 Average Connector Endface edge detection for the operators under Consideration 53
Table 4.2 Multimode Connector ORL measurements 58
Table 4.3 Singlemode Connector ORL measurements 58

LIST OF ABBREVIATIONS

Acronyms	Definition
adps	Average detections per section
dB	decibels
DUT	Device under test
EIA	Electronic Industries Alliance
FOTP	Fiber optic test procedure
GUI	Graphic User Interface
IEEE	Institute Of Electrical Electronics Engineers
IUT	Image under Test
LASER	Light Amplification by Stimulated Emission of Radiation
LED	Light Emitting Diode
LoG	Laplacian of Gaussian
MSE	Mean square error
MM	Multimode
MAE	Mean Absolute Error
MQJ	Measurement Quality Jumper
OCWR	Optical Continuous Wave Reflectometry
OLTS	Optical Loss Test Set
ORD	Optical Reflection Discrimination
ORL	Optical Return Loss
OSNR	Optical Signal to Noise Ratio
OTDR	Optical Time Domain Reflectometer
SM	Singlemode
TIA	Telecommunications Industry Association
VCSEL	Vertical Cavity Surface Emitting Laser

ABSTRACT

In this dissertation we developed a fiber optic connector optical return loss (ORL) model from the connector endface image profile of a 200X inspection microscope. The connector endface profiles were histogram equalized and then evaluated with the Canny, Sobel, Prewitt, Laplacian of the Gaussian (LoG), Roberts and Frei-Chen edge detection operators using Matlab® 2013 script, these operators yielded edge pixel count averages of 139.1, 49.43, 49.57, 103.2, 45.19 and 50.98 detections per section respectively. The Canny operator emerged the best candidate technique after yielding the maximum edge pixel count of 139.1 detections per section, where the figure of merit is the magnitude of the total number of edges detected. The image segmentation and image similarity measure of the Image under Test (IUT) and the Measurement Quality Jumper (MQJ) were implemented using the cross correlation coefficient and pixel based segmentation method respectively so as to establish the coefficients of the connector profile region reflectance in the ORL model. These tools were implemented using a combination of the Matlab 2013 Graphic User interface (GUI), image processing toolbox and symbolic math toolbox. The model results showed a mean connector ORL accuracy of ±0.5dB and ±0.55dB for the multimode and Singlemode connector ORL respectively. These fall within the TIA/EIA-455-95-A maximum specification of ±1dB and ±5dB for the Singlemode and Multimode optical fibers.

CHAPTER ONE

INTRODUCTION

1.1 Background.

Communication can be said to be one of the activities that is as old as human existence itself. It has metamorphosed incredibly over the course of documented human history; from spoken words, sounds, hand gestures, smoke signals etc. All through this evolution, the communication model has remained largely the same with modifications made to each functional block as shown in Figure 1.1 to leverage on prevailing technological advances.

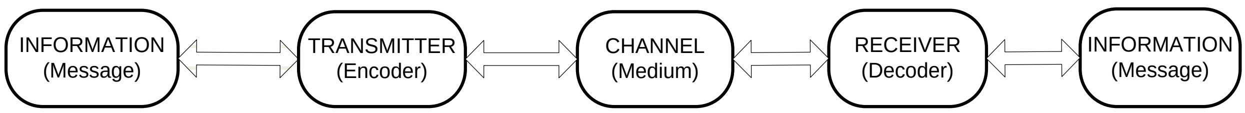


Figure 1.1**:** Generic Communication Network Model (James, 2004)

The use of light for telecommunication dates back to 1790 when a French engineer named Claude Chappe designed one of the most practical applications of light in communication called the optical telegraph (James, 2004). The optical telegraph was used to relay signals over a distance of 230km in about fifteen (15) minutes. However the optical telegraph was soon replaced by the copper telegraph. Researchers, however, remained undaunted in their pursuit of developing an optical communication system and this resolve later paid off with the development of the LASER (light amplification by stimulated emission of radiation) by an American physicist named Theodore Maiman which earned him the appellation of "Father of Electro-optical Industry"(Amandeep *et al.*, 2013). There has been a steady growth in the global adoption of fiber optics due to its large capacity, immunity to electromagnetic interference, attenuation etc (Casimer, 2002).

Nigeria as a country has also not been left out in the adoption of fiber optic technology following investments in the South Atlantic 3/West Africa Submarine Cable (SAT3/WASC) in Lagos by the Nigerian Telecommunication Ltd (NITEL) and installation of a host of backbone fibre rings in the cities of Lagos, Abuja and Kaduna etc.(Vivien, 2011). The SAT3/WASC 120Gbits/second undersea cable provides a communications pathway between Asia and Europe (Philip, 2011). Other privately owned undersea/ terrestrial fiber optic cables have since been deployed such as the GLO-1, Main ONE and the terrestrial 10 Gigabit Campus Fiber Network of the Ahmadu Bello University, Zaria.

1.1.1 Introduction to Optical Fiber Communications System.

Optical fiber communications involves the propagation of modulated light along a fiber optic wave guide based on the principle of total internal reflection. The modulation takes place at the transmitter segment as shown in Figure 1.2 where the electrical signal is converted to light pulses by either Light Emitting Diodes (LED) or Light Amplification by Stimulated Emission of Radiation (LASERS) and demodulated at the receiver by photo detectors.

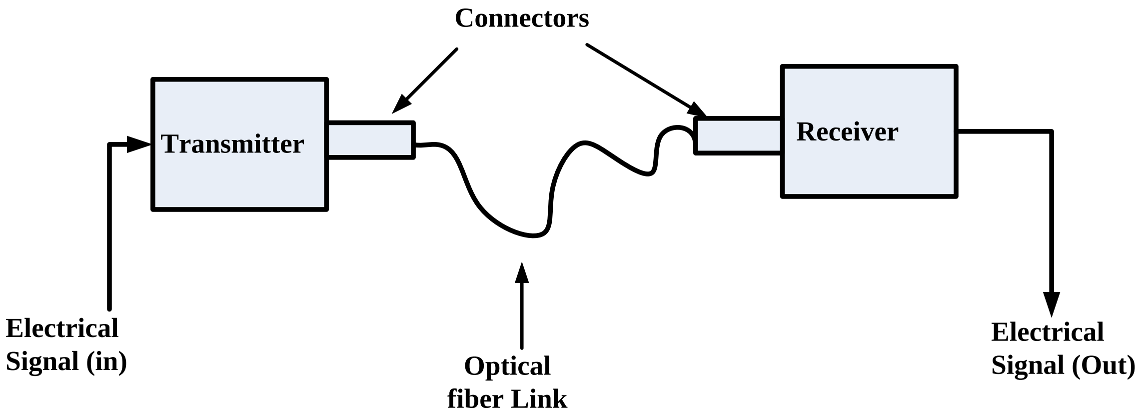


Figure 1.2: A Typical Fiber Optic Link (Bill & Emile, 2005).

As the light pulses propagate, they are susceptible to back reflections or return loss. Laser-diode transmitters offer higher performance than LEDs, but they are also more sensitive than LEDs to light reflected back into them from the fiber optic communication system. The reflected light can change the wavelength of the transmitting laser and add noise to the transmitted signal (destructive interference). Reflections that enter a Vertical Cavity Surface Emitting Laser (VCSEL) affect lasing action in the cavity and add noise to the optical signal (Bill & Emile, 2005). The added noise introduces interference to the fiber optic link, thus leading to data loss and reduced information throughput (Berdinskikh *et al*., 2002).

1.1.2 Digital Image Operations.

An image is a 2-D representation of a three-dimensional scene. This 2-D representation is implemented by a two-dimensional function f(x, y) that represents a measure of some characteristics such as brightness or colour of viewed scene. This connotes that for each position (x, y) in the projection plane, f(x, y) defines the light intensity at that point (Jayaraman *et al.*, 2011)

The manipulation of this 2-D function can be divided into three categories (Ian, 1998):

i. Image Processing: image in $\rightarrow$ image out
ii. Image Analysis: image in $\rightarrow$ measurements out
iii. Image Understanding: image in $\rightarrow$ high-level description out

Image processing refers to those operations that can be applied to digital images to transform an input image a [m, n] into an output image b [m, n]. Where "m" and "n" are integer coordinates with $\{m=0,1,2,\ldots,M-1\}$columns and $\{n=0,1,2,\ldots,N-1\}$ rows, the intersection of a row and column is termed a pixel. These operations are classified into three categories namely (Ian, 1998):

Point Operation: This is an operation in which the value b[m, n] at a specific coordinate in the output image is dependent only on the input value at that same corresponding coordinate a[m, n] in the input image as illustrated in Figure 1.3.

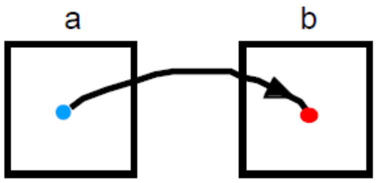


Figure 1.3: Illustration of Point Operation.(Ian, 1998)

Local Operation: An operation is local when the output value at a specific coordinate b[m, n] is dependent on the input values in the neighborhood of that same coordinate a[m, n] in the input image as illustrated in Figure 1.4.

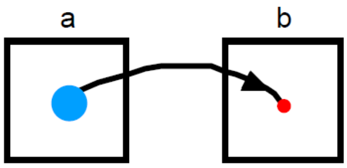


Figure 1.4: Illustration of Local Operation (Ian, 1998).

Global operation: A global image processing operation is one in which the output value at a specific coordinate b[m, n] is dependent on all the values in the input image as illustrated in Figure 1.5.

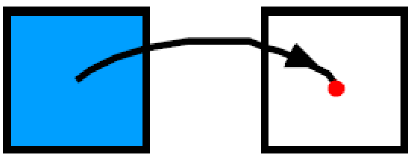


Figure 1.5: Illustration of Global Operation (Ian, 1998).

Image analysis on the other hand investigates the image data to gain insight into the happenings within the image and apply this knowledge to specific applications. The process requires the use of tools such as (Qiang and Roberts, 2015);

i. Image segmentation

ii. Image transformation

iii. Feature extraction and pattern classification.

Segmentation of gray scale images into regions for measurement or recognition is the most important single problem area for image analysis, however a vast array of techniques have been developed in implementing image transformation, feature extraction and pattern classification (John, 2011).

1.2 Aim and Objectives

This research aims at developing a grahic user interface (GUI) based method of measuring optical fiber connector Optical Return Loss (ORL) by exploiting the connector image profile of an inspection microscope. The objectives of the research are as follows:

i. Using the Matlab image processing toolbox, evaluate the performance of edge detection methods in identifying the edges on a connector end-face image profile.

ii. Deduce the mathematical relationship between the edges detected and the connector optical return loss and develop a Matlab graphical user interface [GUI] based application using the established relationship to measure connector return loss.

iii. Validate the results obtained using the measurement quality jumper (MQJ) as a reference.

1.3 Statement of Problem

It has been established from literature that multimode and singlemode fiber optic cables are susceptible to Rayleigh scattering (accounting for about 90% of total attenuation) which occurs when light collides with individual atoms in the glass knocking it off its original course (Bagad, 2009) while absorption (which accounts for between 3%-5% of total attenuation) is caused by light being absorbed by residual materials such as metals or water ions within the fiber core and inner cladding (Vivek, 2004). The Rayleigh scattering and absorption induced attenuation are manufacturer dependent (Vivek, 2004). Return loss falls in the category of the remaining 5%-7% contribution to overall attenuation.

This dissertation seeks to extend the functionality of a typical fiber inspection microscope to not only give a visual display of the endface of fiber optic connector but also compute the connector optical return loss (ORL) which will enable the engineer to make objective assessment of the suitability of a field terminated optic fiber cable.

This is achieved by identifying and implementing using Matlab GUI, image processing toolbox and symbolic math toolbox the appropriate edge detection method, image segmentation and similarity measure to determine the connector optical return loss (ORL) from the connector endface image profile of a fiber inspection microscope thereby improving upon its contemporary function of visual inspection alone.

1.4 Methodology

The procedures implemented for this dissertation are:

i. Termination of three (3) sets each of both multimode and singlemode optical fiber cables using ST connectors.
ii. Using the 200X fiber optic inspection microscopes, insert the terminated fiber optic cable in step (i) into the microscope and capture the end-face image of the connector.
iii. Using an OTDR capture the reflective event trace and record the optical return loss for the multimode and singlemode fiber optic connectors respectively.

iv. Using the Matlab image processing toolbox and the captured end-face image profile from step (ii) transform these profiles with the variants of the following edge detection methods:

   a. Canny method
   b. Sobel method
   c. Prewitt method
   d. Roberts method
   e. Laplacian of Gaussian method
   f. Frei-Chen method

v. Plot and evaluate the output of these transforms so as to select the method that yields the most edges as the candidate method.

vi. Transform the connector end-face image profiles with the candidate method in step (v) and establish the relationship between the transformed image profile and the optical return loss measured in step (iii) above using the Matlab image processing toolbox commands.

vii. Develop a MATLAB Graphic User Interface application that uses the established relationship in step (vi) to perform measurements on new connector end-face samples.

viii. Validate the results from the algorithm to be developed by comparing it with that of the measurement quality jumpers (MQJ)

1.5 Dissertation Outline

The general introduction has been presented in chapter one. The other chapters are outlined as follows; the review of fundamental concepts of image processing and analysis closely followed by the review of similar research works was presented in chapter two. The edge detection variants were implemented and evaluated using MATLAB 2013 software in chapter three. In Chapter four the results of the edge detection operators and validation of the developed connector ORL model is presented and discussed. The conclusions drawn, limitations encountered, significant contributions and recommendations for further work were covered in chapter five.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter is split into two major sections namely the review of fundamental concepts and the review of similar works in order to establish the basis for the tools used in this dissertation.

2.2 Review of Fundamental Concepts

This sub-section presents a review of the fundamental concepts of this research.

2.2.1 Wave Propagation.

The propagation of optical fields in optical fibers is governed by Maxwell's equation for a non-conducting medium without free charges are represented by (Govind, 2002):

$$\vec{\nabla} \times \vec{E} = -{}^{\partial \vec{B}}/_{\partial t} \tag{2.1}$$

$$\vec{\nabla} \times \vec{H} = -{}^{\partial \vec{D}}/_{\delta t} \tag{2.2}$$

$$\vec{\nabla}.\vec{D} = 0 \tag{2.3}$$

$$\vec{\nabla}.\vec{B} = 0 \tag{2.4}$$

where,
$\vec{\boldsymbol{E}}$ and $\vec{\boldsymbol{H}}$ are the electric and magnetic field vectors,

and $\vec{\boldsymbol{D}}$ and $\vec{\boldsymbol{B}}$ are the corresponding flux densities.

The flux densities are related to the field vectors by the constitutive relation in (Govind, 2002):

$$\vec{D} = \epsilon_o \vec{E} + \vec{P} \tag{2.5}$$

$$\vec{B} = \mu_o \vec{H} + \vec{M} \tag{2.6}$$

where

ϵ_o is vacuum permittivity,
μ_o is vacuum permeability
and $\vec{\boldsymbol{P}}$ and $\vec{\boldsymbol{M}}$ are the induced electric and magnetic polarizations.

However, optical fibers do not have magnetic polarization due to the non-magnetic nature of silica glass, as shown in equation (2.7):

$$\mathbf{M=0}. \qquad (2.7)$$

The optical field wave equation is therefore expressed mathematically as (Govind, 2002):

$$\nabla \times \nabla \times \mathrm{E} = \frac{-1}{\mathrm{c}^2}\,\frac{\partial^2 \mathrm{E}}{\partial^2 \mathrm{E}^2} - \mu_o \frac{\partial^2 \mathrm{P}}{\partial \mathrm{t}^2} \qquad (2.8)$$

2.2.2 Optical Fiber profiles and modes.

An optical mode refers to a specific solution of the wave equation that satisfies the appropriate boundary conditions and has the property that its spatial distribution does not change with propagation. There are three types of fiber modes namely guided modes, leaky modes and radiation modes (Bagad, 2009).

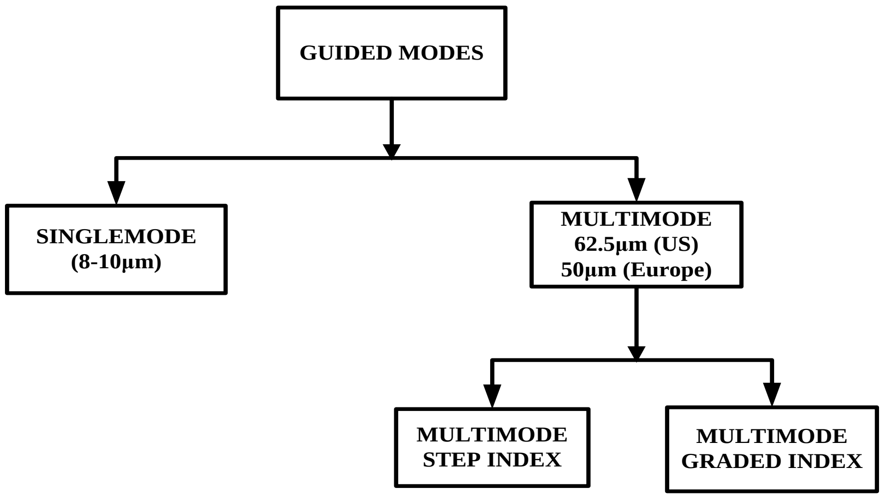


Figure 2.1: Classification of Optical Guided Modes. (Pierre, 2008)

However, fiber optic communication systems employ the use of guided modes for the propagation of light pulses. Guided modes are further classified based on the refractive index profiles of their cores (Bagad, 2009), the classification are as shown in Figure 2.1

2.2.3 Optical fiber attenuation

As light pulses propagate through an optical fiber, it experiences a decrease in power level, This decrease in power level is expressed in dB or as a rate of loss per unit distance (dB/km). The two main loss mechanisms of light transmission in optical fiber are light absorption and scattering.

Absorption Loss: All materials absorb light at wavelengths corresponding to resonance frequencies in the molecules. The light absorption occurs when a photon excites an electron to a higher energy level. This phenomenon obeys urbachs empirical rule which is represented mathematically as (Ivan, 2013):

$$\alpha_a = A\,exp\left(\frac{B}{\lambda}\right) \tag{2.9}$$

where

α_a is the absorption loss

λ is the wavelength of the light pulse

A and B are inherent constants of the material used

Scattering Loss: Scattering loss arise from refractive index variations caused by microscopic heterogeneities such as fluctuations in material density or composition. Raman and Brillouin scattering is produced as a result of temperature and strain variations along the fiber optic link (Ivan, 2013). The refractive index variations smaller than the wavelength of the light pulse induces Rayleigh scattering is expressed mathematically as (Ivan, 2013)

$$\alpha_s = 10\log_e\left(\frac{8\times\pi^3\times n^8\times p^2\times K_B\times T_f\times\beta_T}{3\times\lambda^4}\right) \tag{2.10}$$

where;

n is the refractive index

p is the photoelastic coefficient

K_B is the Boltzmann's constant

T_f is the fictive temperature at which the density fluctuations are frozen

β_T is the isothermal compressibility of the material

λ is the wavelength of the light pulse

The optical power attenuation due to Rayleigh scattering is expressed mathematically in equation (2.10) (Ivan, 2013).

2.2.4 Optical Return Loss (ORL)

Optical return loss (ORL) is the logarithmic ratio of the launch (incident) power divided by the total reflected power seen at the launch point. The total reflected power is the total accumulated reflected optical power measured at the launch caused by fiber Rayleigh scattering and Fresnel reflections. Rayleigh scattering is the scattering of light along the entire length of the fiber, caused by elastic collisions between the light wave and fiber molecules. This results in some of the light being reflected back to the source. Rayleigh scattering is intrinsic to the fiber and therefore cannot be eliminated (Noutsios, 2006)

Fresnel reflections occur in the light path where there is an abrupt change in the refractive index such as at connectors and splices. The further away a reflective event is from the fiber launch point the less it contributes to the total reflected power. Therefore, fiber connectors and splices closest to the laser contribute the most to the ORL.

ORL is always expressed as a positive decibel. The higher the ORL the lower the reflected power. Optical return loss expressed in dB in equation (2.11) is defined as the logarithmic ratio of incident power to the total received power back at the source caused by all components of the system including the fiber (Pierre, 2008):

$$ORL = 10 \times \log\left(\frac{P_{inc}}{P_{refl}}\right) \quad \geq 0 \tag{2.11}$$

where

ORL is optical return loss in decibels (dB)

P_{refl} is the optical power reflected back towards the transmitting source of energy measured at the input of the device,

P_{inc} is the optical power in the forward direction measured at the input of the device,

$P_i is\ not\ equal\ to P_r$

A low ORL effectively reduces the modulation bandwidth of the laser and signifies the presence of multiple reflections within the optical communication system which is observed by an increase in Bit Error Rate (BER) for digital systems and diminishes Signal to Noise Ratio (SNR) in analogue systems (Mohammad *et al.,* 2010).

Several groups have established minimum performance standards for return loss in fiber-optic communication systems. The TIA/EIA-568A1 series of standards, established jointly by the Telecommunications Industry Association and the Electronic Industry Association, specifies a fiber optic network return-loss value greater than 26dB for fiber-optic premises networks.

There are several methods that can be used to measure ORL. The most common methods are (Rongquing & Maurice, 2009):

i. Optical Continuous Wave Reflectometry (OCWR),

ii. Optical Time Domain Reflectometry (OTDR), and

iii. Optical Reflection Discrimination (ORD).

2.2.4.1 *Optical Continuous-Wave Reflectometry*

The most widely used method for measuring return loss is OCWR. Using this method, a single wavelength of light is launched at a known power level into the fiber-optic system and to the device under test. The chosen wavelength should be close to the wavelength the communication system will use in its intended application. A directional coupler routes back reflections to a detector in an optical power meter as shown in Figure2.2.

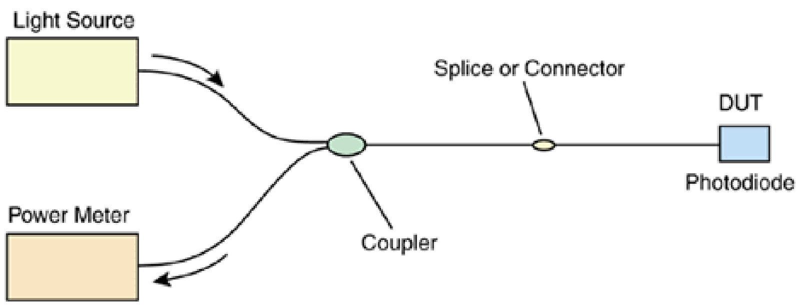


Figure 2.2: A Simple OCWR Measurement Setup (Laferrière *et al.,* 2007).

The measurement system can be calibrated using the Fresnel reflection from the fiber-to-air interface at a standard polished connector. Such a connector used in place of the device undergoing testing, provides a reflection of –14.7 dB at 1550 nm (Rongquing & Maurice, 2009). The connector is then substituted with the device under test (DUT) and the difference in returned power and the return loss is determined.

Although the OCWR method is simple, it has a few drawbacks. The power meter will measure power from all reflections that occur in a fiber, not just those reflections from a Device Under Test (DUT). The DUT in this case is the fiber optic connector. The reflections that do not come from the DUT can amount to more than the reflections from the DUT, thus limiting the sensitivity of the power meter. A “mandrel wrap” or a termination plug is required at the end of the fiber to eliminate reflections from components other than the DUT.

The OCWR technique requires the use of a directional coupler that would not be affected by different modes of light in a multimode fiber-optic system. Unfortunately, multimode signals do affect the performance of common fused-construction couplers. The coupling ratio in these devices changes depending on the distribution of the modes. The mode distribution in turn depends on the characteristics of the laser and any connections between it and the coupler. In addition, modal distribution depends on temperature, so during testing, ambient temperature has to be controlled (Rongquing & Maurice, 2009).

Using an LED as a light source for OCWR measurements may cause problems because the spectrum of LED is more broader than that of a laser. Furthermore, the LED’s spectrum does not match that of the laser that will produce the signal actually used for communications. The LED also produces less power. Thus, using an LED as the light source will affect measurement accuracy.

The limitations of OCWR stems from the fact that power meters respond to all reflections and the characteristics of directional-couplers depend on the wavelength and mode of light transmissions. Tests require a mandrel wrap or similar termination at the far end of a cable.

2.2.4.2 *Optical Time Domain Reflectometer (OTDR)*

An optical time-domain reflectometer can also be used to measure optical return loss. An optical time-domain reflectometer (OTDR) launches light pulses into the device under test and then collects backscatter information and superimposed Fresnel reflections. The OTDR computes and displays loss (decibels) against unit distance (meters) based on the backscatter signals received by the instrument.

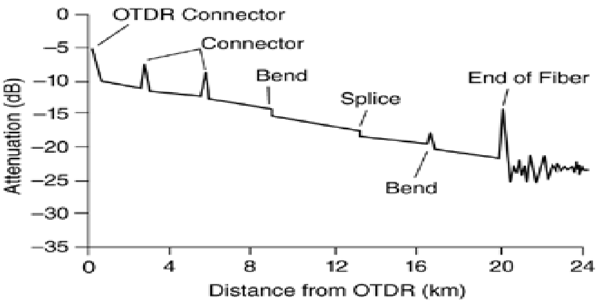


Figure 2.3: OTDR Trace of Power Loss against Distance from the Launch Point (Bill, 2005).

The display plots points on a graph; the vertical axis represents relative energy in a reflection and the horizontal axis represents the location of the reflecting component. The horizontal axis represents time as shown in Figure 2.3, but users preset the speed of light in the fiber undergoing testing so the OTDR can convert time to distance and display distance on the horizontal axis. The TIA/EIA-455-8 series of documents contain procedures for measuring splice or connector loss and reflectance using an OTDR.

In order to yield accurate results when measuring return loss using an OTDR, the DUT must be connected beyond the instrument's "dead zone" (which is the distance over which the photodetector in an OTDR is saturated momentarily after it measures a strong reflection).

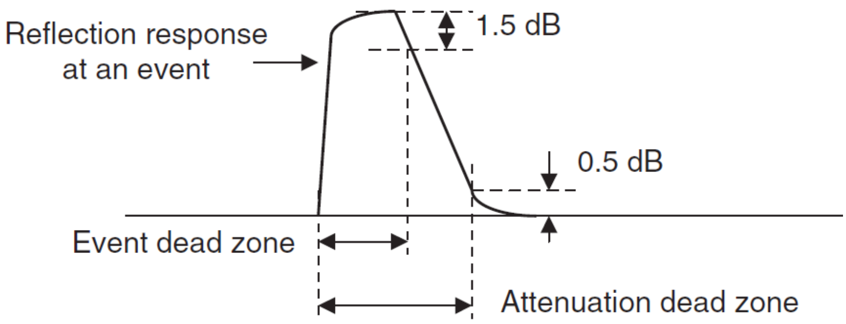


Figure 2.4: Event and Attenuation Dead Zone (Gerd, 2006).

The actual dead-zone length varies by OTDR make and model, but it is usually a few meters. The dead zone length represents the distance over which the OTDR cannot make measurements because reflections in this length saturate the OTDR. Thus, reflections that occur in the dead zone cannot be detected or appear on an OTDR trace as a vague anomaly instead of as discrete pulses as shown in Figure 2.4. Typical OTDRs have an event dead zone (EDZ) of 1m for singlemode fibers and 0.2m for multimode fibers (Mohammad *et al.*, 2010). An OTDR can take up to 30 seconds to make a series of measurements and calculate a result.

2.2.4.3 *Optical Reflection Discrimination*

The third measurement technique uses Optical Reflection Discrimination (ORD) to separate backscatter from actual reflections. ORD works in much the same way as optical time-domain reflectometry. A stable laser source injects light pulses into the fiber and thus into the DUT. But the ORD technique uses pulse widths chosen to minimize backscatter. In general, a pulse width of 10 nm or less results in a sufficiently strong signal that would not cause significant backscatter. An ORD instrument exists as a single piece of equipment, although an ambitious test engineer could configure an ORD test setup using a laser pulser, a coupler and a sensitive optoelectric converter *(*Laferrière *et al.*, 2007).

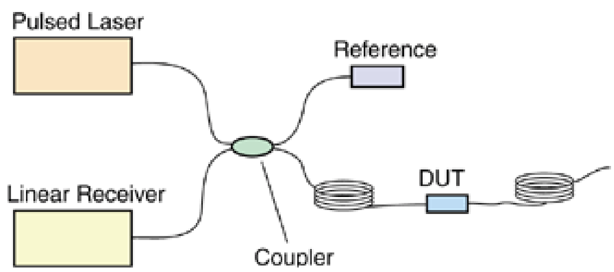


Figure 2.5: Setup for ORD Measurements *(*Laferrière *et al.*, 2007).

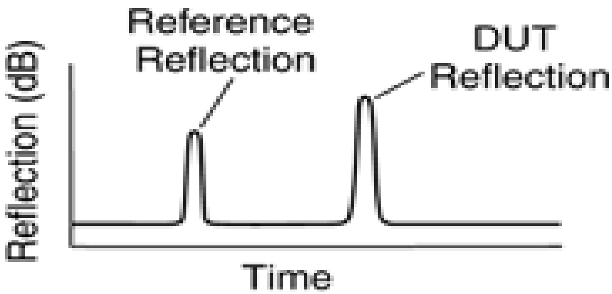


Figure 2.6: Sample ORD Measurement Plot (Pierre, 2008)

Like an OTDR, an ORD instrument guides reflected light through a directional coupler to a separate receiver as shown in Figure 2.5. However, unlike an OTDR, an ORD instrument uses a time-discrimination process implemented in hardware and software to capture only certain signals. The discriminator captures signals that rise above a minimum level and signals with a “fast” slope. In this way, the discriminator filters out the backscatter and captures only reflections. The receiver converts the light to an electrical signal, amplifies the signal, and measures it. The ORD instrument records a time base and can plot the amplitude of the reflections and the time of their arrival as in Figure 2.6. The built-in software analyzes the reflection signal and reports the return loss of the DUT to the user. An ORD instrument incorporates a highly linear optical receiver; thus, the amplitude of measured reflections is directly proportional to the return loss.

The edges or discontinuities that account for the connector return loss can be observed from the output of a fiber inspection microscope. In order to accurately identify these edges and compute the resultant return loss image processing tools have to deployed before measurements can be made off the image.

2.2.5 Edge Detection

Edge detection in image processing involves the identification of points at which the image brightness experiences discontinuities; these discontinuities are a consequence of geometric or non-geometric events (Trucco & Verri, 1998).

Detecting these edges and gleaning pertinent information from the image profile requires the use of an appropriate operator, the ideal operator will depend on a set of criterion, which include:

i. edge orientation
ii. noise environment and
iii. edge structure

Edge detection operators are classified into two broad categories (Raman & Himanshu, 2003)

i. Gradient based methods
ii. Laplacian based methods

2.2.5.1 Gradient based Methods

The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. For a one (1)-dimensional function f(x) the derivative takes on values with a large magnitude at the points where the function has a high rate of change. It is reasonable to expect that these points will correspond to edges in the function as shown in (Baghai, 2013).

$$f'(x) = \frac{df}{dx} \tag{2.12}$$

The generalization of f(x) to a 2-D function f(x, y) is the gradient expressed mathematically as (Baghai, 2013):

$$\nabla f(x, y) = \left(\frac{\partial f\,(x,y)}{\partial x} i_x + \frac{\partial f\,(x,y)}{\partial x} i_y\right) \tag{2.13}$$

Where i_x and i_y are unit vectors in the x and y directions respectively.

The magnitude of $f(x, y)$ is first computed, and is then compared to a threshold to find candidate edge points.

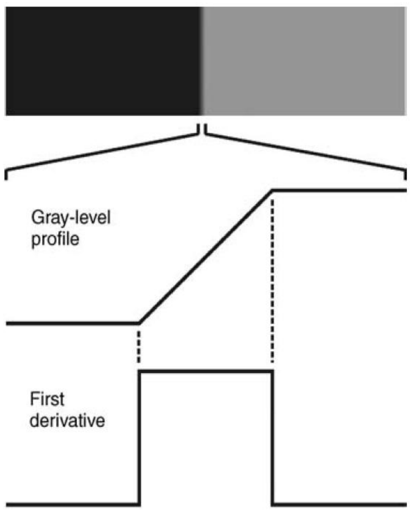


Figure 2.7: Intensity Profile and First Derivative Results.(Haldo and Juan, 2015).

A pixel location is declared an edge location if the pixel intensity of the local minima or maxima of the first derivative exceeds predetermined threshold, edges will have higher pixel intensity values than those surrounding it as shown in Figure 2.7. So once a threshold is preset, the gradient value is compared to the threshold value and an edge is detected when the threshold intensity is exceeded (Trucco & Verri, 1998).

Edges are positions in the image where the image function changes, however, since an image is a 2D function, these positions are identified using the gradient of the function $\nabla f(x, y)$.

Discrete signal processing is handled by the use of difference equations or a convolution sum. This implies that numerical differentiation involves taking the running difference of a discrete signal as expressed in (Patrick, 2004):

$$y(t) = \frac{dx(t)}{dt} \equiv y[n] = x[n] - x[n-1] \tag{2.14}$$

The gradient of a 2-D function is a 2-D vector with components the partial derivatives of the function along two orthogonal directions. In the discrete case these partial derivatives are the partial differences computed along two orthogonal directions by using masks. (Maria & Panagiota, 1999)

There are many possible derivative approximation filters for use in gradient estimation. Two as (Wufeng *et al*., 2014):

First difference:

$$f(n) - f(n-1) \tag{2.15}$$

Central difference:

$$\frac{1}{2}[f(n+1) - f(n-1)] \tag{2.16}$$

These result in the following 1-D convolution filter as represented by:

First difference:

$$h(n) = \delta(n) - \delta(n-1) = [1 - 1] \tag{2.17}$$

Central difference:

$$h(n) = \delta(n+1) - \delta(n-1) = \frac{1}{2}[1\ 0 - 1] \tag{2.18}$$

Thus the difference operations described can be viewed as the convolution of f(n1; n2) with the impulse response of a filter $h(n_1; n_2)$. Each impulse response has h_1 corresponding to a filter for detecting horizontal edges, and h_2 for detecting vertical edges (Baghai, 2013).

The variants of Gradient based operators include Prewitt, Sobel, Roberts and Canny operator e.t.c (Rashmi *et al.*, 2013).

2.2.5.2 Lapalacian based Edge Detection

This method searches for zero crossings in the second derivative of the image profile to find edges. The Laplacian is applied to an image that has first been smoothed with an operator approximating a Gaussian Smoothing filter in order to reduce its sensitivity to noise (Baghai, 2013). In effect, the Gaussian operator reduces the noise and Laplacian operator detects the sharp edges (Rashmi, 2013).

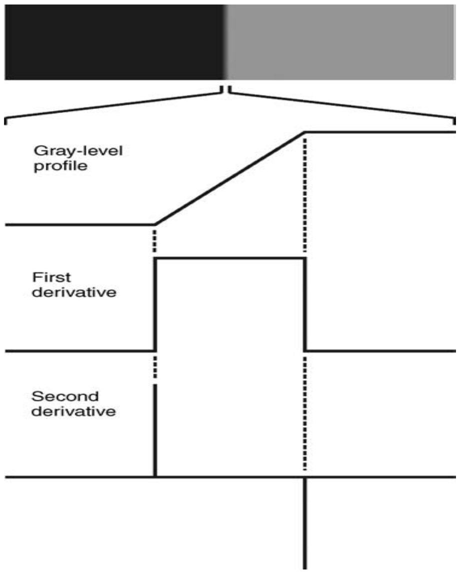


Figure 2.8: Intensity Profile: First and Second Derivative Results (Marques, 2011).

The points which lie on the zero-crossings of the second derivative are candidate edge points as shown in Figure 2.8, the pixel intensity of these candidate points are compared against a threshold to isolate the true edges (Trucco & Verri, 1998).

The Gaussian function is expressed mathematically as (Rashmi *et al.*, 2013):

$$G(i,j) = \frac{1}{\sqrt{2\times\pi\sigma^2}}\exp^{-\left(\frac{i^2+j^2}{2\times\sigma^2}\right)} \tag{2.19}$$

where

σ is the standard deviation.

The Laplacian of Gaussian operator is then computed mathematically as (Rashmi *et al.*, 2013):

$$LoG = \frac{\partial^2}{\partial i^2}G(i,j) + \frac{\partial^2}{\partial j^2}G(i,j) = \frac{i^2+j^2-2\sigma^2}{\sigma^4}exp\left(-\frac{i^2+j^2}{2\sigma^2}\right) \tag{2.20}$$

The classical variants of these edge detection schemes are as follows (Haldo and Juan, 2015):

i. Roberts operator

Roberts operator's component filters employ first difference equations and are tuned for diagonal edges rather than vertical and horizontal ones. The correlation masks are given by (Baghai, 2013):

$$h_1 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \tag{2.21}$$

$$h_2 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.22}$$

Where,
h_1 and h_2 are the vertical and horizontal convolution kernels respectively.

The pair of Robert filters has a common zero crossing point for their differencing kernels. Edge location must be assigned to actual pixel location, namely the one at the filters origin. Due to its reliance on first difference equation Roberts operators have high sensitivity to noise which can be minimized by incorporating smoothing into each filter in the direction normal to that of the difference.

ii. Prewitt

The Prewitt edge gradient operator simultaneously accomplishes differentiation in one direction using the central difference and noise reduction in the orthogonal direction by means of local averaging. Its use of central difference accounts for its lesser edge-location bias. The Prewitt filters h_1 and h_2 which corresponds to horizontal and vertical edges detecting filter are expressed as (Baghai, 2013):

$$h_1 = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \tag{2.23}$$

$$h_2 = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \tag{2.24}$$

where h_1 and h_2 are the vertical and horizontal convolution kernels respectively.

However, the Prewitt operator is less sensitive to diagonal edges than to vertical or horizontal ones.

iii. Sobel

This operator is one of the most widely used gradient edge detectors and often a better option than Prewitt because the low-pass filter produced by the $[1 \quad 2 \quad 1]$ kernel results in a smoother frequency response, although it employs the center difference equation. However, it is more sensitive to diagonal edges than to vertical or horizontal ones. The kernels for this filter are expressed as (Baghai, 2013):

$$h_1 = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \tag{2.25}$$

$$h_2 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \tag{2.26}$$

where h_1 and h_2 are the vertical and horizontal convolution kernels respectively.

iv. Frei-Chen Operator

The need for a gradient based operator with an equal gradient magnitude response to diagonal and vertical edges birthed the Frei-chen operator. The Frei-Chen filters h_1 and h_2 which corresponds to horizontal and vertical edges detecting filter are expressed as (Wufeng *et al.*, 2014):

$$h_1 = \begin{pmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{pmatrix}, \tag{2.27}$$

$$h_2 = \begin{pmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{pmatrix} \tag{2.28}$$

where h_1 and h_2 are the vertical and horizontal convolution kernels respectively. However the Frei-Chen operator retains some directional sensitivity in gradient magnitude, making it not truly isotropic because the difference operators used are not rotationally symmetric.

v. Laplacian of Gaussian Edge Detection (LoG)

This non-directional method (devised to tackle both vertical and horizontal edges (Li and Jean, 2013) searches for zero crossings in the second derivative of the image profile to find edges. The image is first smoothed with an operator approximating a Gaussian Smoothing filter before applying the Laplacian derivative operation due to its sensitivity to noise (Baghai, 2013). The standard deviation in the Gaussian distribution function controls the degree of noise filtering therefore a larger value of the standard deviation may blur the image inadvertently causing edge boundaries to be lost. The mathematical expression of the LoG kernel is (Haldo and Juan, 2015):

$$LoG(i,j) = \frac{1}{\pi\sigma^4}\left(1 - \frac{(i^2 - j^2)}{2\sigma^2}\right) e^{-\left(\frac{i^2+j^2}{2\sigma^2}\right)} \qquad (2.29)$$

Where

σ is the standard deviation of the image pixel intensities

i and j are integers 1,2,3, . . . representing the coordinate of pixel location.

vi. Canny operator

This operator was developed for optimal detection of edges in images containing white noise (Khan and Ravi, 2013). It is implemented in three phases which are (Tranos *et. al.,* 2014):

i. Smoothening and differentiation
ii. Non-maximal suppression
iii. Thresholding

i. Smoothing and Differentiation.

In this phase the input image is smoothened with a Gaussian to obtain the gradients in the ‘x’ and ‘y’ directions. The image gradient which shows changes in the intensity indicates the presence of edges.

ii. Non-maximal suppression

Since the edges occur at points where the gradient is at maximum, it means all points not at maximum should be suppressed. This is achieved by analyzing each pixel; if the magnitude of the gradient is greater at one pixel's distance in either the positive or negative direction perpendicular to the gradient. If the pixel is not greater than both, suppress it.

iii. Thresholding

This phase of the Canny operator uses hysteresis to implement its thresholding, the fate of each ridge pixel is then determined according to the following criteria (Chris and Toby, 2011):

i. if $|G(x,y)|< T_1$, then the pixel is rejected and is not an edge pixel;

ii. if$|G(x,y)|>T_2$, then the pixel is accepted and is an edge pixel;

iii. if T1 < $|G(x,y)|< < T_2$, the pixel is rejected, except where a path consisting of edge pixels connects it to an unconditional edge pixel with $|E(x,y)| > T_2$.

where:

T_1 and T_2 are predefined thresholds,

and $|G(x,y)|$ is pixel magnitude.

2.2.6 Image Convolution theorem

Convolution is a widely used mathematical operator that processes an image by computing for each pixel a weighted sum of the values of that pixel and its neighbours one of the most important results of Fourier theory is that the convolution of two functions in real space is the same as the product of their respective Fourier transforms in Fourier space. In practice this is expressed mathematically as (Jayaraman*et al.*, 2011):

$$g(x, y) = \sum_{k=-n_2}^{n_2} \sum_{j=-m_2}^{m_2} h(j, k) \cdot f(x - j, y - k) \tag{2.30}$$

where;

m_2 is equal to half of the masks width

n_2 is equal to half of the masks height and

x is the floor operator, which rounds a number to the nearest integer less than or equal to x.

2.2.7 Image Histogram

The histogram of an image represents the relative frequency of occurrence of the various gray levels in an image (Maria & Panagiota, 1999). For a digital image with a range of gray levels, the histogram represents a discrete function expressed mathematically as (Tinku & Ajoy, 2005):

$$\boldsymbol{P(r_k) = \frac{n_k}{N}} \tag{2.31}$$

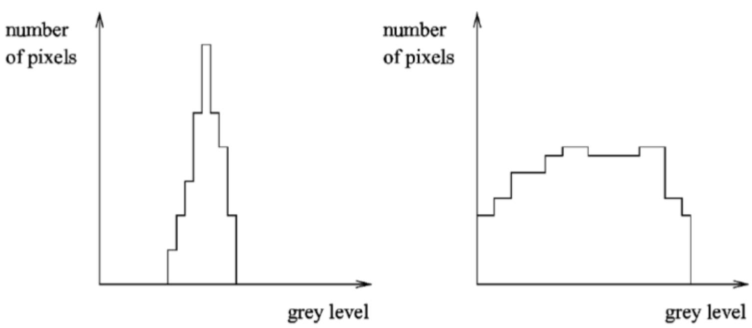
where

k is from 0,1,…,L-1

r_k Is the k^{th} gray level

n_k is the number of pixels in the image with that gray level

N is the total number of pixels in the image.



(a) Poor Image Histogram (b) Good image Histogram

Figure 2.9: Image Histograms (Maria & Panagiota, 1999).

The significance of the image histogram is that a narrow histogram means the details in an image are sparse and not suitable for gleaning pertinent information while an image with a broad histogram connotes a detailed image as depicted in Figure 2.9.

However an image with a narrow histogram can be remedied by histogram equalization which consists of adjusting the gray scale of the image so that the gray level histogram of the input image is mapped unto a uniform histogram (Tinku & Ajoy, 2005)

2.2.8 Image Segmentation

Segmentation is defined as the process of partitioning an image into a set of non-overlapping regions whose union is the entire image. These regions should ideally correspond to objects and their meaningful parts, and background. Most image segmentation algorithms are based on one of two basic properties that can be extracted from pixel values discontinuity and similarity or a combination of them (Khan and Ravi, 2013).

The goal of image segmentation is to cluster pixels into salient image regions, i.e., regions corresponding to individual surfaces, objects, or natural parts of objects. Segmentation could be used for object recognition, occlusion boundary estimation within motion or stereo systems, image compression, image editing, or image database look-up (Mitchell, 2010).

The input of a segmentation block in a machine vision system is a pre-processed image, whereas the output is a representation of the regions within that image. This representation can take the form of the boundaries among those regions (e.g., when edge-based segmentation techniques are used) or information about which pixel belongs to which region (e.g., in clustering-based segmentation). Once an image has been segmented, the resulting individual regions (or objects) can be described, represented, analyzed, and classified appropriately. The different segmentation methods are classified into the following categories.

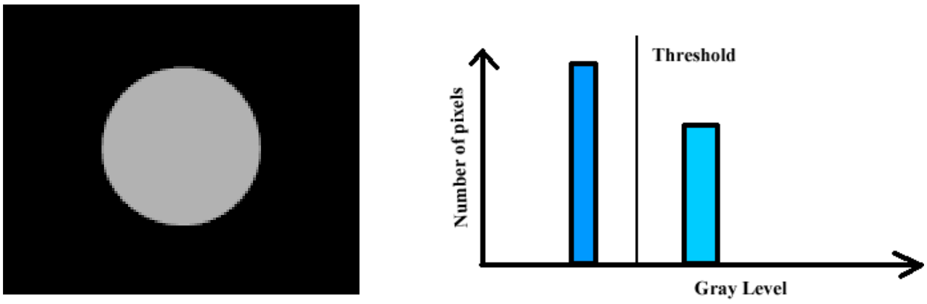
2.2.8.1 Pixel-Based Segmentation

Point-based or pixel-based segmentation is conceptually the simplest approach for implementing image segmentation as it leverages solely on the colour features at individual pixels while ignoring spatial constraints making it susceptible to noise in images. This accounts for the presence of isolated small regions that are not present in noise free images (Tzafestas, 2001).

The simplest technique of the pixel based segmentation is histogram thresholding (the histogram is often a valuable tool in establishing a suitable threshold Value). When several desired segments in an image can be distinguished by their grey values, threshold segmentation can be extended to use multiple thresholds to segment an image into more than two segments: all pixels with a value smaller than the first threshold are assigned to segment zero, all pixels with values between the first and second threshold are assigned to segment one, all pixels with values between the second and third threshold are assigned to segment two, *etc*. If there are *n* thresholds (t_1, $t_{2,\ldots}t_n$) are used as shown in (Fritz, 2011):

$$g(v) = \begin{cases} 0 & \text{if} \quad v < t_1 \\ 1 & \text{if} \quad t_1 \leqslant v < t_2 \\ 2 & \text{if} \quad t_2 \leqslant v < t_3 \\ \vdots & \vdots \quad \vdots \\ n & \text{if} \quad t_n \leqslant v. \end{cases} \tag{2.32}$$

After thresholding, the image has been segmented into *n*+1 segments identified by the grey values 0 to *n* respectively as illustrated in Figure 2.10.



(a) Sample Image (b) Histogram of Sample Image

Figure 2.10: Illustration of Pixel Based Segmentation (John, 2011)

Clustering is another pixel-based technique that is used for image segmentation. In this technique, a histogram is first obtained by the colour values at all pixels and the shape of each cluster is found. Then each pixel in the image is assigned to the cluster that is closest to the pixel colour.

2.2.8.2 *Edge based segmentation*

Detects and links edge pixels to form contours. They are sensitive to texture variations and impulsive noise. Since a (binary) object is fully represented by its edges, the segmentation of an image into separate objects can be achieved by finding the edges of those objects (Fritz, 2011).

Algorithm for edge-based segmentation

Given an image *f*,

i. Compute an edgeness image ∇f from *f*. Any preferred gradient operator can be used.
ii. Threshold ∇f to an image $(\nabla f)\ t$, so we have a binary image showing edge pixels.
iii. Compute a Laplacian image Δf from *f*. Any preferred discrete or continuous Laplacian operator may be used.
iv. Compute the image using (Fritz, 2011):

$$g = (\nabla f)\ t \cdot \operatorname{sgn}(\Delta f) \qquad (2.33)$$

2.2.8.3 *Region based segmentation*

Segmenting an image into regions is directly accomplished through region-based segmentation which makes it the most popular technique currently in use. This technique considers both colour distribution in colour space and spatial constraints (John, 2011). In theory, finding an object by locating its boundary and finding it by establishing the region it covers will gives exactly the same object; the boundary and the region are just different representations of the same object. Region based segmentation methods have only two basic operations: splitting and merging and many methods even feature only one of these. The basic approach to image segmentation using merging is:

i. obtain an initial (over) segmentation of the image,
ii. merge those adjacent segments that are similar –in some respect– to form single
iii. segments,
iv. go to step 2 until no segments that should be merged remain.

The basic form of image segmentation using splitting is:

i. obtain an initial segmentation of the image,
ii. Split each segment that is inhomogeneous in some respect (*i.e.*, each segment that is unlikely to really be a single segment).
iii. go to step (ii) until all segments are homogeneous.

In general Region based methods are robust because Regions cover more pixels than edges and thus more information is available in order to characterize the region of interest.

__2.2.8.4__ Model based segmentation

This are stochastic model based techniques in which the image regions are modelled as random fields and the segmentation problem is posed as a statistical optimization problem if the exact shape of the objects in the image is known. This technique often provides more precise characterization of the image regions because it is not restricted to local image information. Some of the spatial interaction models used are Markov random Fields (MRF) or Gibbs random Field (GRF) to model digital images. Model based techniques tend to be computationally intensive and thus are only used in cases where complexity is not a problem (John, 2011).

2.2.9 Image Similarity Assessment

In other to ascertain the similarity between two images a number of measures have been developed. They however differ depending on the representation of the cases. Image similarity measures can be generally categorized into the following groups (Qiang & Robert, 2015):

i. Feature-based measures
ii. Pixel-based measures and
iii. Structure-based measures

__2.2.9.1__ Feature-based Measures

This technique expresses image similarity as a measure of distance d(r, s) between the reference image “R” and the corresponding image under test “I” i.e. to find the offset (r, s) such that the similarity between the shifted reference image R(r, s) and ”I” is a maximum as shown in Figure 2.10 (Fritz, 2011).

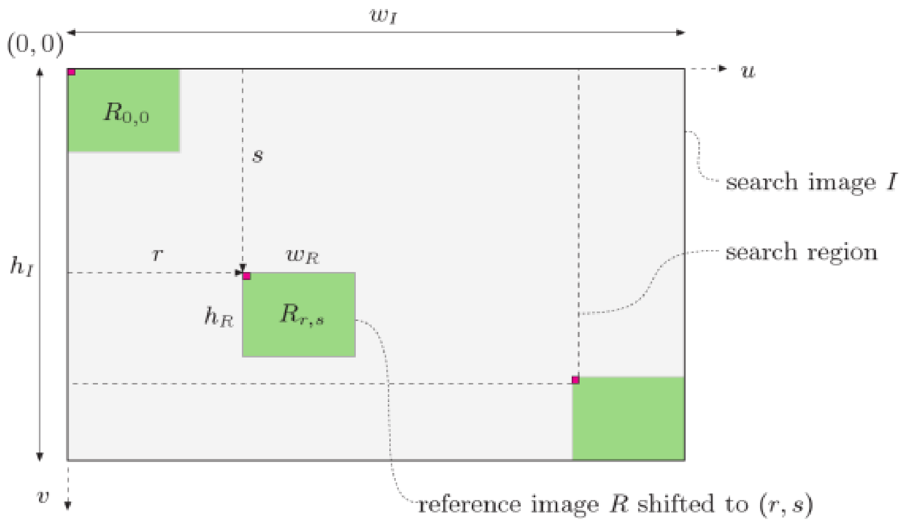


Figure 2.11: Illustration of Feature Based Similarity Measure (Fritz, 2011).

Where $(i, j) \in R$

$0 \leq i < w_R; 0 \leq j < h_R$

I is the search image

R is the reference image

$w_R \rightarrow widthofreference\ image$

$h_R \rightarrow heightofreference\ image$

(r, s) vertical and horizontal position.

Several distance measures for two-dimensional images exist in literature however the basic measures are (Quiang & Roberts, 2015):

i. Sum of absolute differences
ii. Maximum difference
iii. Sum of squared difference

i. Sum of Absolute Difference (SAD)

This measure is implemented by computing the difference within a square neighbourhood of the reference image (R) and the image under test (I). This is followed by the aggregation of the absolute differences. An exact match is denoted by a zero resultant. This measure is expressed mathematically as (Quiang & Roberts, 2015):

$$d_A(r,s) = \sum_{(i,j)\in R} \left| I_{(r+i,s+j)} - R_{(i,j)} \right| \tag{2.34}$$

where (i, j)∈ R

$0 \leq i < w_R; 0 \leq j < h_R$

$w_R \rightarrow widthofreferenceimage$

$h_R \rightarrow heightofreferenceimage$

(r, s) vertical and horizontal position.

Furthermore, it has been established from literature that SAD distance measures are strongly affected by global intensity changes (Annadurai, 2007).

ii. Maximum Difference

This measure is an empirical process implemented by computing the difference between the Image under test (I) and the reference image at various offset positions. This is followed by finding the maximum of the absolute values. It has been established from literature that this measure is completely useless as a distance measure since it responds more strongly to the lighting changes than to pattern similarity (Annadurai, 2007). This measure is expressed mathematically as (Mitchell, 2010);

$$d_M(r,s) = max_{(i,j)\in R} \left| I_{(r+i,s+j)} - R_{(i,j)} \right| \tag{2.35}$$

Where (i, j)$\in R$

$0 \le i < w_R; 0 \le j < h_R$

$w_R \rightarrow widthofreferenceimage$

$h_R \rightarrow heightofreferenceimage$

(r, s) vertical and horizontal position.

iii. Sum of Squared Differences.

This measure is called the N-dimensional Euclidean distance with 'N' being the number of pixels used in the distance computation. The sum of the square of the differences between the values in the reference image to that in the image under test (I) is expressed mathematically as (Mitchell, 2010):

$$d_E(r,s) = \left[\sum_{(i,j)\in R}\left(I_{(r+i,s+j)} - R_{(i,j)}\right)^2\right]^{1/2} \tag{2.36}$$

where (i , j)$\in R$

$0 \le i < w_R; 0 \le j < h_R$

$w_R \rightarrow widthofreferenceimage$

$h_R \rightarrow heightofreferenceimage$

(r, s) vertical and horizontal position.

The Euclidean distance $d_E(r,s)$ expression can be expanded as seen in

$$d_E^2(r,s) = \underbrace{\sum_{(i,j)\in R} I_{(r+i,s+j)}^2}_{A(r,s)} + \underbrace{\sum_{(i,j)\in R} R_{(i,j)}^2}_{B} - \underbrace{2 \times \sum_{(i,j)\in R} I_{(r+i,s+j)} . R_{(i,j)}}_{C(r,s)} \tag{2.37}$$

$A_{(r,s)}$depends only on the filter and is the same for every image pixel, B is the sum of squares of pixel values in the reference image which is a constant value and $C_{(r,s)}$ is the linear cross correlation between I and R. The Euclidean distance is inversely proportional to the similarity of the reference and test image.

2.2.9.2 *Pixel-based Measures*

This measure operates on the raw image data and is therefore featureless since they do not impose any level of image abstraction. They are easy to compute and quite effective in content-based image retrieval. Two fundamental featureless image similarity measures are (Qiang and Robert, 2015):

i. Normalized Cross-correlation
ii. Cross correlation coefficient

i. Normalized cross-correlation

The normalized cross-correlation compensates for the dependency in Sum of Squared Differences (SSD) by taking into account the energy in the reference and test images making it less susceptible to intensity changes in the test image. It is expressed mathematically as (Jayaraman, *et al*., 2011):

$$C_{N(r,s)} = \frac{\sum_{(i,j)\in R} I_{(r+i,s+j)} R_{(i,j)}}{\left[\sum_{(i,j)\in R} I^2_{(r+i,s+j)}\right]^{\frac{1}{2}} \sum_{(i,j)\in R} R^2_{(i,j)}} \tag{2.38}$$

If the values in R and I are all positive like in gray scale images then the result is always in the range [0, 1] with $C_{N(r,s)} = 1$ denoting a maximum match.

ii. Cross-Correlation Coefficient.

This measure is used when the images are captured by the same sensor and any changes in illumination may be approximated with a linear transformation. It is expressed mathematically in the spatial domain as the sum of the products of the pixel brightness divided by their geometric mean as expressed in equation (2.39) (Raghavender *et al.*, 2014).

$$C_{L_{(r,s)}} = \frac{\sum_{(i,j)\in R}\left(I_{(r+i,s+j)} \times R_{(i,j)}\right)}{\sqrt{\sum_{(i,j)\in R} I^2_{(r+i,s+j)} \times \sum_{(i,j)\in R} R^2_{(i,j)}}} \tag{2.39}$$

When the dimensions of the summation are large this is a slow and inefficient process compared to the equivalent operation in frequency space which is expressed mathematically as shown in equation (2.40) (Raghavender *et al.*, 2014):

$$C_{(u,v)} = I_{(u,v)} * R_{(u,v)} \tag{2.40}$$

where $*$ indicates the complex conjugate of the function values.

This is also referred to as a dyadic operation because two images are involved and not a filter or mask (which is a frequency domain image).

__2.2.9.3__ Structure- Based Measure.

This measure is well suited in comparison of global images which contain multiple intersecting structures and for which the spatial relationship of these structures is important. The simplest methods of structure-based measures are of two types (Mitchell, 2010):

i. Mean Square Error (MSE)

This is a statistical measure, which sums the squares of the difference in the gray-level intensity pixel value of the test image with that of the reference image at corresponding pixel positions; squaring is done so that negative values do not cancel positive values during summation (Ajit and Dorothy, 2000) following which an arithmetic mean is computed as shown in equation (2.41):

$$MSE = \sum_k \frac{(I_k - R_k)^2}{k} \qquad (2.41)$$

Where,

I_k and R_k are the gray-levels of the k^{th} pixel of the reference and test images.

The greater the computed MSE value the lower the image similarity and vice versa. MSE image similarity measure is poor when used to measured image similarity across distortion types however, it possesses lower computational complexities compared with human visual system models (Eric et al., 2007).

ii. Mean Absolute Error (MAE)

The MAE is also a statistical measure that computes image similarity as a function of the absolute value in gray-level pixel value in corresponding pixel positions of both test and reference images following which an arithmetic mean is computed as:

$$MAE = \sum_k \frac{|I_k - R_k|}{k} \qquad (2.42)$$

Where,

I_k and R_k are the gray-levels of the k^{th} pixel of the reference and test images.

High MAE values connote low image similarity (Eric et al., 2007).

2.3 Review of Similar Works

A total of thirteen (13) similar works were reviewed and are summarized as follows:

Berdinskikh *et al.*, (2002) studied the influence of contamination and scratches on the Bit-Error-Rate (BER) of ST and SC connectors and established a strong inter relation. The mathematical relationship between the end face contamination and BER was not concluded thereby limiting the repeatability of computations. Furthermore, the analysis was not extended to low form factor connectors thereby limiting its application.

Zuyuan *et al.*, (2004) developed a novel model for analyzing the optical interface performance degradation due to scratches on optical fiber end face. The model indicated that the contribution to the Return Loss (RL) of a scratch was determined by its size, location, and its relative reflectivity, which was defined as the ratio of the average reflectivity of the scratch to the base reflectivity of the defect less end-face. This methodology however involves measurement of the length and width of each defect off the end face image profile which was laborious and prone to errors thereby unsuitable for return loss measurement in field terminated connectors, furthermore it does not check for concentricity of the core-cladding boundaries.

Mai *et al.*, (2004) proposed methods to evaluate Optical Time Domain Reflectometer (OTDR) and Optical Return Loss Meter (ORLM) for field applications. Variable reflectance references for multimode and single mode fibers were built to evaluate the attenuation dead zones of OTDRs. Evaluation of the OTDR attenuation dead zone against a reflectance event of -40 dB with recovery to within 0.5 dB of backscatter reflection was discussed. A variable reflection standard was built to evaluate ORLM models from three different manufacturers in the range of [10 dB to 45 dB] of return loss with an accuracy of +/-0.5 dB. The standard was verified down to -45 dB return loss within +/-0.5 dB. Testing showed that equipment adapter and fiber connector geometries, fiber contact cleanliness, and skill were also critical to the evaluation. However, this method does not guarantee repeatability of measurement (the ability to obtain consistent results when measuring the same part with the same measuring instrument).

Francesco, *et al*., (2010) established that the most severe limiting factor in densely integrated photonics was backscattering and derived a general relationship between backscattering and the geometrical optical parameters of the optical wave guide. The drawback however was that physical observation which is highly subjective was used to estimate the backscatter thereby resulting in poor repeatability performance of the method.

Junzhi, *et al*., (2010) explored the use of edge detection algorithms in road centreline detection and estimation of road widths which were vital to many autonomous driving systems. Canny transform was used to detect the edge of a line, and then the parallel lines were extracted from the edges through progressive probabilistic Hough transform. The experimental results showed that this methodology was effective in estimating road width and further served as a scientific basis for the use of edge detection schemes in parameter estimation.

Lu *et al*., (2011) developed a segmentation scheme which isolates human forms using a 2 –D contour model and 3-D head surface model from depth information taken by the kinect Xbox 360. The method presented consistent results however, the kinect Xbox 360 not being a visible-light camera fell short as it did not consider images from contemporary visible-light images.

Selim, *et al.*, (2011) developed a model to efficiently predict and solve abnormal conditions in electrical cables, using Matlab GUI (Graphic User Interface) with thermal imaging (infrared (IR) camera). This model detects anomalies in electric power cables by analyzing the thermal signature and comparing it against normal working conditions thermal signature. Although the model applied only to electric power cables and not to fiber optic cables it provided a scientific reference for adaptable procedures for gleaning measurements from image profiles of device under test.

Yajima *et al.*, (2011) investigated experimentally the performance of field assembly connectors using incorrectly cleaved fiber ends. The parametric quantities used were the insertion and return loss which measured 40dB and 30dB respectively. The Optical Time Domain Reflectometer (OTDR) was used which inadvertently included the backscatter along the fiber. Furthermore it exposes the laser of the OTDR to degradation of lasing action which is imminent in the presence of high reflectance.

Koyama *et al.*, (2012) developed a physical contact (PC) connection by using simple butt joining between an SC connector and a raw cleaved Singlemode (SM) fiber end without index matching gel. A crack depth of less than 5µm and 100mm bending radius the optimum values in conventional fiber cleaving were adhered to which yielded an average insertion and return loss of 0.14dB and 51.1dB respectively. The repeatability of this approach and the applicability to angular physical contact (APC) was not considered.

An-Tsung et al., (2013) experimentally established the efficacy of using image partitions referred to as sub-images in histogram-equalized images as against the deployment of global transform functions. The edge preserving rate was improved and therefore served as a scientific basis for the use of sub-images in image analysis functions.

Kihara *et al.*, (2013) investigated the performance of physical contact (PC) fiber optic connectors with various air-filled gaps. It was discovered experimentally that both insertion and return loss of imperfect PC connectors deteriorates by 9.4dB and 9.9dB respectively. However the mathematical relationship for this deterioration was never established for application on subsequent imperfect PC connectors. The research did not however consider the relationship between the imperfections of contemporary angled physical contacts (APC) and the losses produced.

Quiang *et al.*, (2013) implemented a comparison of the measurement principle and application of the Optical Continuous Wave Reflectometry (OCWR) and the Optical Reflection Discrimination (ORD) methods in determination of the optical return loss (ORL) of the connectors, couplers, switches and detects all components of a fiber optic communication system so as to mitigate the system back reflection which results in high bit error rate as a result of multipath interference, degraded optical signal-to-noise-ratio (OSNR) and transmitter instability. OCWR was found to have improved measurement accuracy while the ORD has the advantage of simplicity of implementation due to its Mandrel wrap free operation and increased dynamic range. Furthermore, OCWR was found to be more efficacious in measuring ORL of an entire link while ORD fared better for quick ORL measurement of individual optical devices. The ORD technique however compromises accuracy for speed which underscores the need for a technique that embodies the advantages of the OCWR and ORD.

Shikama *et al.*, (2013) demonstrated with return loss as the parametric basis the effect of the physical contact (PC) multicore connector conditions. A return loss of 40dB was established on all cores however the effect on angular physical contact was not considered thereby limiting the scope of applicability of the approach.

This dissertation deploys a different approach from Zuyuan *et al.*(2004), by using edge detection image processing in MATLAB application to ascertain the parameters (length, width and proximity to the core) of chips, scratches and other defects and in turn use these data to model the connector optical return loss.

CHAPTER THREE

MATERIALS AND METHODS

3.1 INTRODUCTION

In this chapter the procedures adopted for implementation of this dissertation in order to achieve the stated objectives.

3.2 Image Histogram

In order to determine the suitability of the connector endface image profile, an image histogram is plotted. The flow chart of the implementation of the image histogram equalization is as shown in Figure 3.1

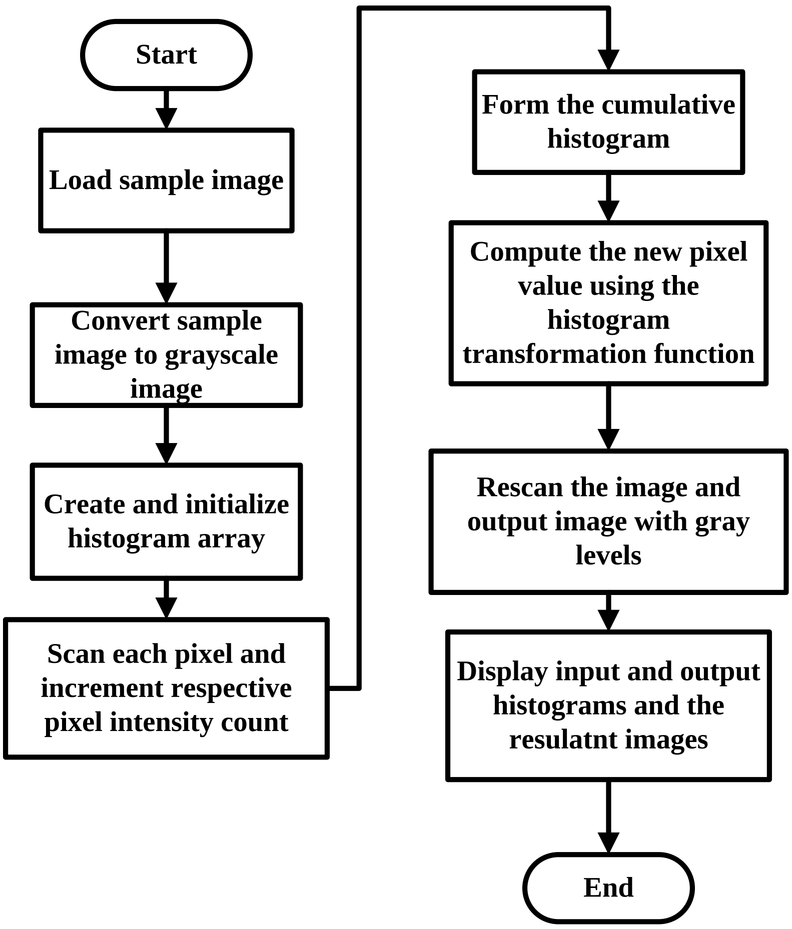
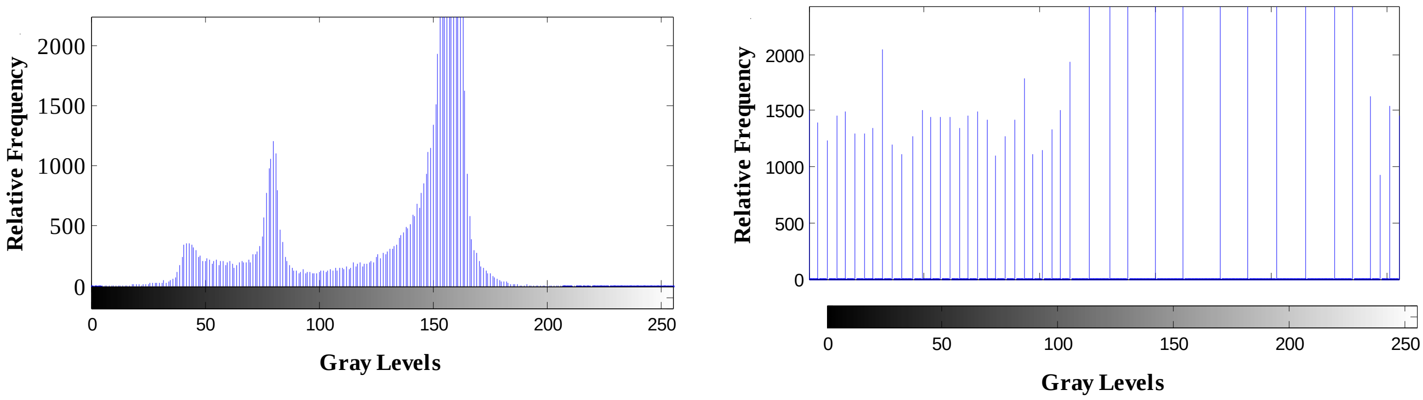


Figure 3.1: Flow Chart of the Image Histogram Equalization Process.

A sample singlemode connector endface profile histogram was shown in Figure 3.2 (See Appendix A for source code). The unequalized gray scale spectrum in Figure 3.2a spans 160 bins (pixel intensity levels) while the equalized gray scale spectrum spans 256 bins Figure 3.2b. This implies that the equalized gray scale spectrum has 256 distinct detectable levels in its image profile which enables it to yield more edges compared to the unequalized image profile which has 160 pixel intensity levels.

Image histogram evaluation and equalization is important to the edge detection schemes because it eliminates noise and make less apparent edges to be detectable.



a) Unequalized Connector Endface Histogram (b) Equalized Connector Endface Histogram

Figure 3.2: Grayscale Connector Endface and Histograms.

3.3 Edge Detection Operators

The classical edge detection operators are analyzed are as follows:

i. Roberts Operator
ii. Prewitt Operator
iii. Sobel operator
iv. Frei-chen Operator
v. Laplacian of Gaussian (LoG)

vi. Canny operator

The flowchart of the edge detection implementation is as shown in Figure 3.3

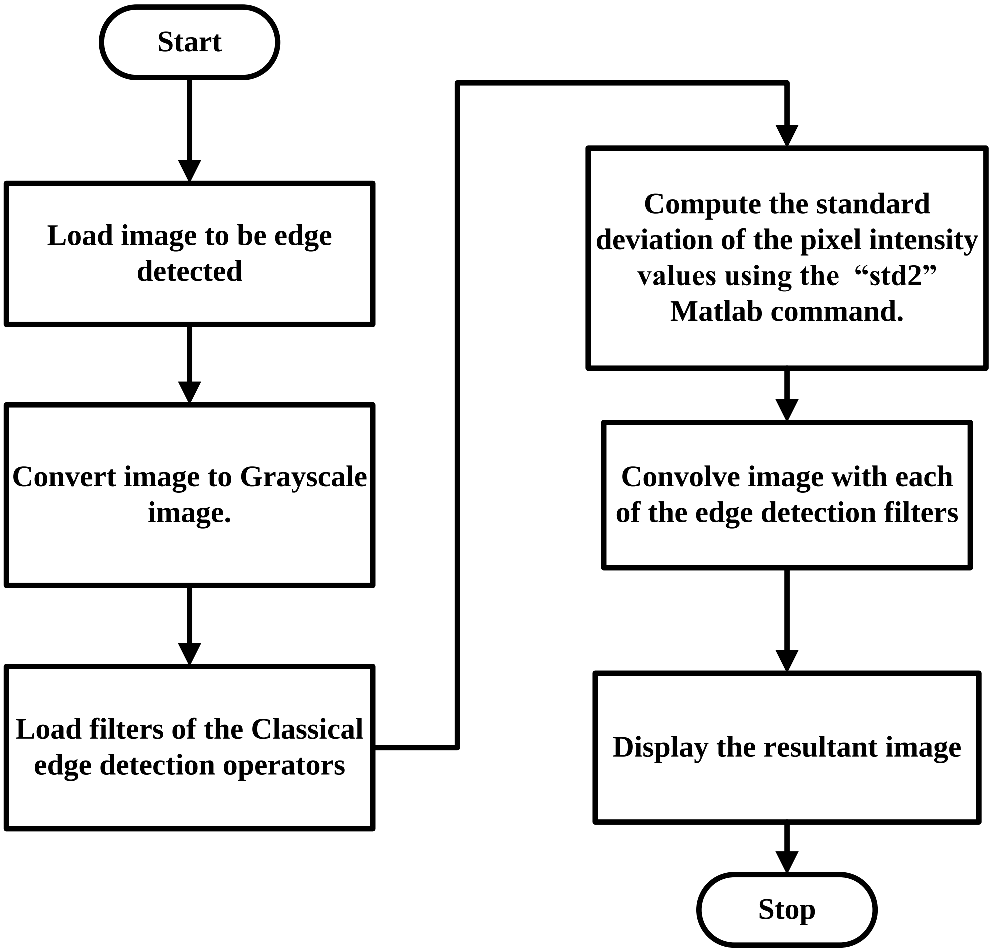


Figure 3.3: Flow Chart of edge detection implementation.

The filter kernel of each of the methods enumerated is convolved with the grey scaled connector image profile to detect the discontinuities in the image.

3.4 Pixel-Based Segmentation.

The pixel based segmentation is implemented so as to accommodate the various contributions of the core, cladding and ferrule surface (Qiang & Roberts, 2015). The flow chart of the algorithm is as shown in Figure 3.12

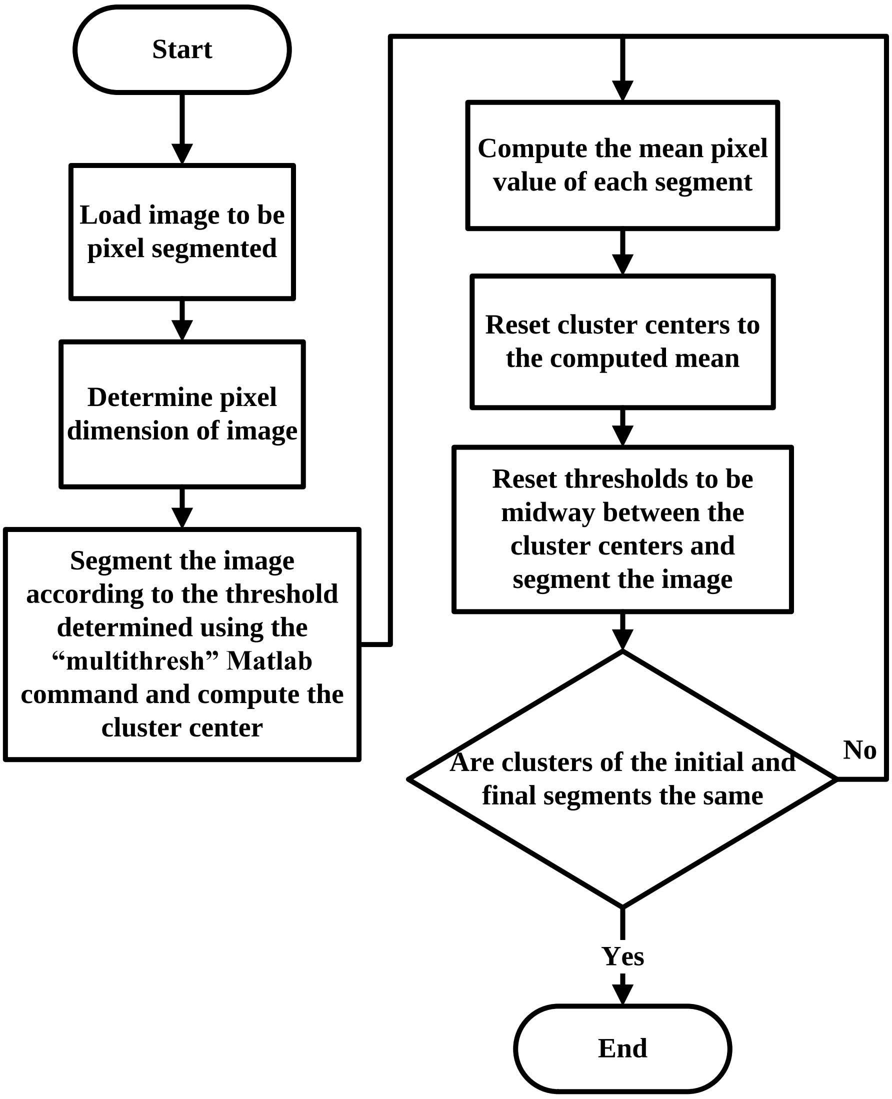


Figure 3.12: Flowchart of Pixel Based Segmentation.

3.5 Development of Connector Endface Return loss model equation.

The connector Endface image profile in Appendix A is loaded and a pixel based segmentation method implemented in Matlab is applied producing three connector Endface segments before a cross correlation coefficient image matching measure is applied on all three segments. This measure is most appropriate (Mitchell, 2010) due to the fact that all the images come from the same sensor which is the fiber optic inspection microscope.

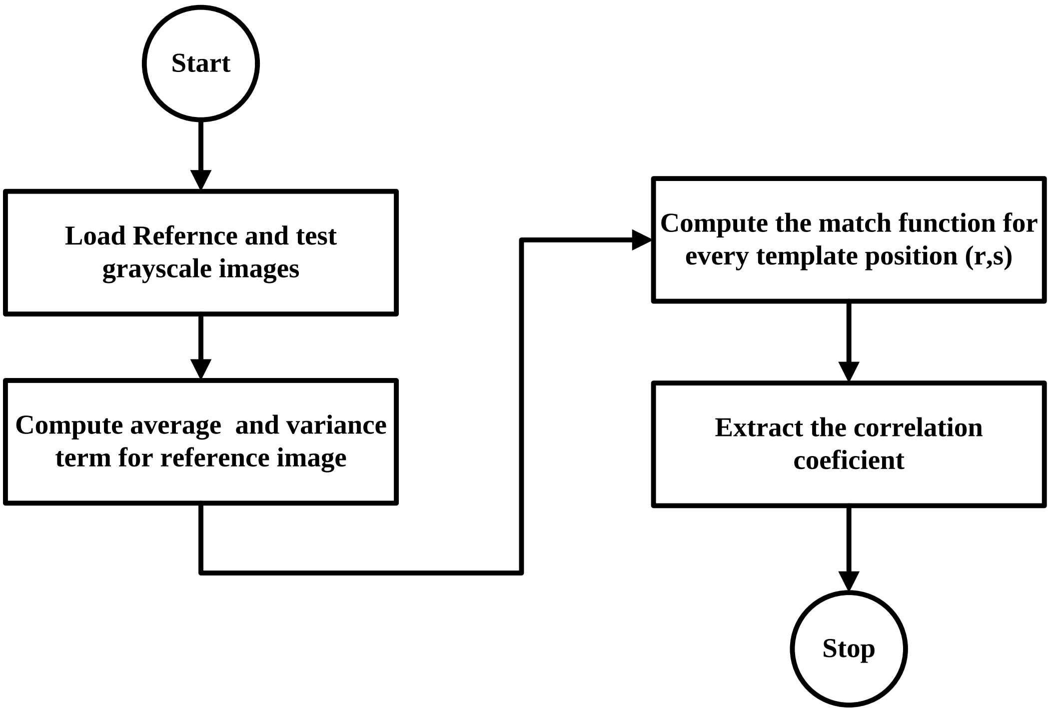


Figure 3.13: Flowchart of Image Similarity Coefficient Extraction.

These coefficients are then used in conjunction with the measured connector return loss to model the return loss equation that will be used in the Matlab Graphic user interface. Matlab source code is on appendix B and the flow chart is shown in Figure 3.13.

CHAPTER FOUR

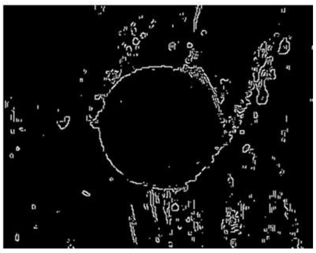
RESULTS AND DISCUSSION

4.1 Introduction

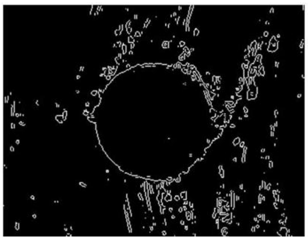
This chapter analyzes the results of the pixel based segmentation and return loss model equation is developed to implement the modelling and simulation of ORL in fiber optic connector endface using an inspection microscope.

4.2 Edge Detection Methods Evaluation

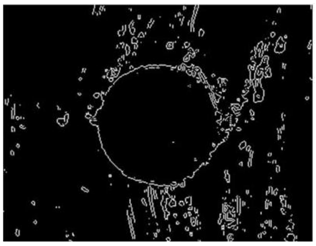
The outputs of the six (6) edge detection techniques were established with their respective outputs shown in Figure 4.1, an quantitative evaluation of the number of detected edges each technique's output was implemented in order to avoid reaching subjective conclusions.



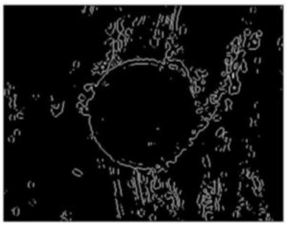
(a) Roberts



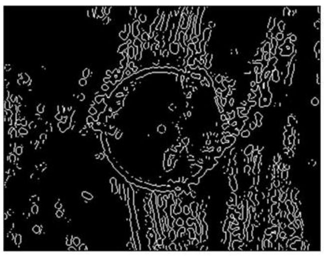
(b) Prewitt



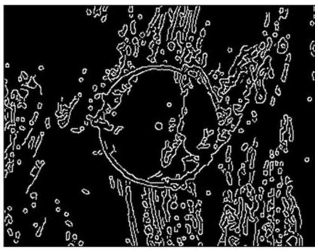
(c) Sobel



(d) Frei-Chen



(e) Laplacian of Gaussian



(f) Canny

Figure 4.1: Graphical Outputs of the Edge Detection Operators. (Source Appendix A)

Due to the close interconnection between edges on a connector endface and its return loss the edge detection technique that yields the most edge counts will be the most suitable candidate technique because it takes into account more discontinuities which produce the reflectance that sum up to constitute the connector return loss. However to ensure consistency of highest number of edge detections, each technique's output profile is split into segments of 5 x 263 pixels each in order to visualize all the segments in a single row (see appendix E) and the total number of edges in each segment is counted. This was implemented with Matlab and the plot of the "sum of detected edges against each output profile segment is shown in Figure 4.2 (See appendix B for source code).

The plot in Figure 4.2 shows that the Canny operator detected more edges across all the image profile segments followed by the Laplacian of the Gaussian (LoG) operator. The other operators which incidentally are gradient based operators detected less edges and all posted similar results evidenced by the overlapping plots in Figure 4.2.

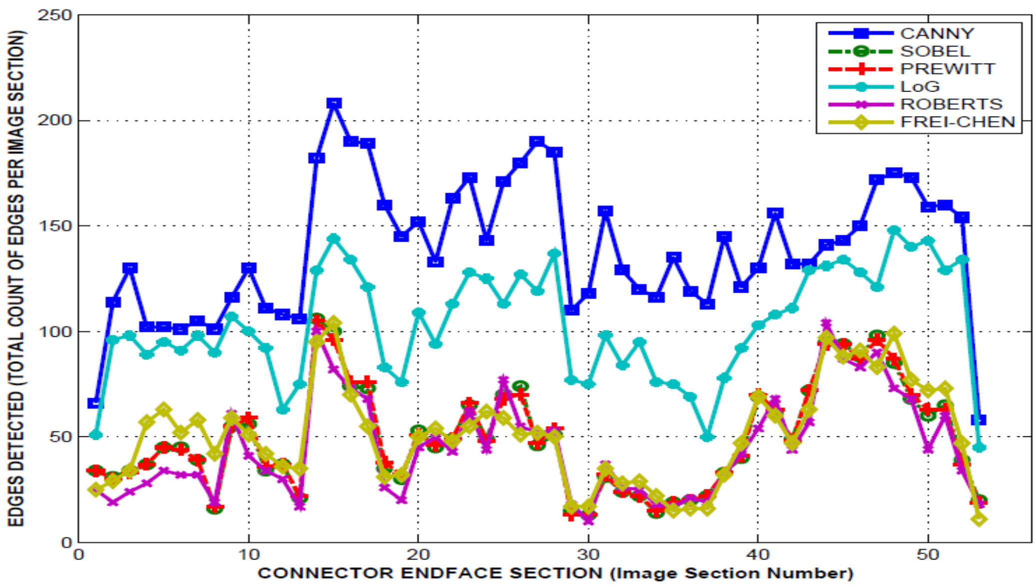


Figure 4.2: Plot of the Sum of Detected Edges along Each Section of the Connector Endface.

To narrow down the empirical evaluation the average number of detected edges across the connector endface profile segments is computed from Appendix C and the results are shown in Table 4.1.

Table 4.1 Average Connector Endface Edge Detection for the Operators under Consideration (Source Appendix C)

Edge Detection Operators	**Canny**	**Sobel**	**Prewitt**	**Laplacian of Gaussian (LoG)**	**Roberts**	**Frei-Chen**
Average No. of Edge Detections	**139.1**	**49.43**	**49.57**	**103.2**	**45.19**	**50.98**

The canny operator yields an average of 139.1 edges per section while the Laplacian of the Gaussian follows closely with 103.2 edges per section. The other techniques, however, have similar average edge detections per section because they are first derivative based methods.

4.3 Fiber optic Connector ORL Model

The coefficients of the representative reflectance of the three (3) pixel based segmentation regions of each of the connector endface (Appendix F) is implemented in Appendix G with the computation of the multimode and singlemode correlation coefficients respectively as follows:

$$Corr\_Coef_{SM} = \begin{bmatrix} -0.0099 & 0.0254 & -0.0461 \\ 0.0509 & 0.0925 & 0.0756 \\ -0.0482 & -0.0037 & 0.0055 \end{bmatrix} \tag{4.1}$$

$$Corr\_Coef_{MM} = \begin{bmatrix} 0.0631 & 0.1104 & 0.0834 \\ 0.1253 & 0.1134 & 0.1142 \\ 0.0673 & 0.0074 & 0.0454 \end{bmatrix} \tag{4.2}$$

Furthermore, the measured connector ORL for the IUT as shown in Appendix F are as follows:

$$Meas\_ORL_{SM} = \begin{bmatrix} 56.6 \\ 55.7 \\ 54.1 \end{bmatrix} \tag{4.3}$$

$$Meas\_ORL_{MM} = \begin{bmatrix} 15.9 \\ 18.1 \\ 24.8 \end{bmatrix} \tag{4.4}$$

Equations 4.1to 4.4 are now combined using the inverse matrix method in MATLAB to compute the coefficients of the connector ORL model with the following syntax:

```
clc;
clear all;
%%%%%%%%  SINGLEMODE   %%%%%%%%%%%
Coeffs_SM=[-0.0099 0.0254 -0.0461;0.0509 0.0925 0.0756;-0.0482 -0.0037 0.0055];
OTDR_RL_SM=[56.6;55.7;54.1];
Ref_Coeff_SM=Coeffs_SM\OTDR_RL_SM
%%%%%%%%%%%%% MULTIMODE %%%%%%%%%%%%
Coeffs_MM=[0.0631 0.1104 0.0834;0.1253 0.1134 0.1142;0.0673 0.0074 0.0454];
OTDR_RL_MM=[15.9;18.1;24.8];
Ref_Coeff_MM=Coeffs_MM\OTDR_RL_MM
```

This syntax yielded the base endface-region return loss for the singlemode and multimode connector ORL model as shown in Figure 4.3.

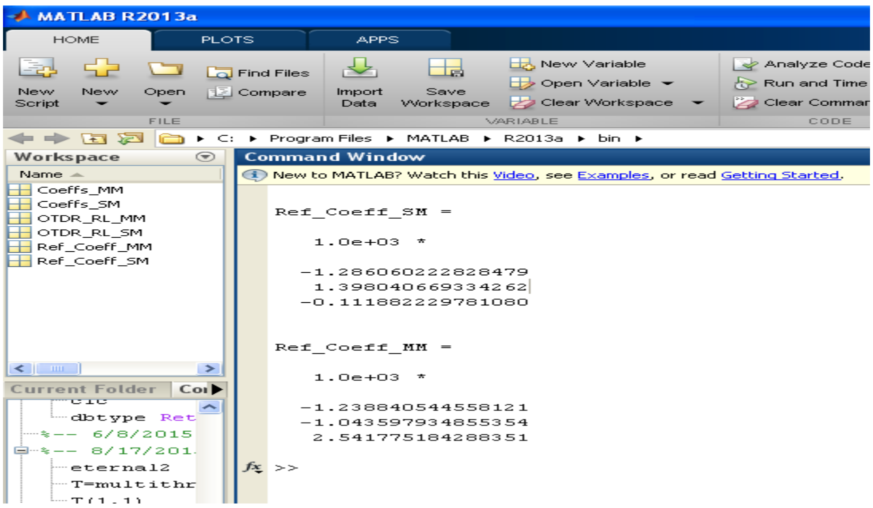


Figure 4.3: Screen Grab of the Matlab Base ORL computation

Using the general syntax of linear equations for n unknowns represented by (Tranos *et al.*, 2014):

$$C_n B_n + C_{n+1} B_{n+1} + C_{n+2} B_{n+2} = R_{cn} \tag{4.5}$$

Where,

n are integers from 1, 2,3.

C_n is the correlation coefficient of image region n,

B_n is the base return loss of pixel region n,

and R_{cn} is the ORL of connector n.

The connector ORL for the singlemode and multimode are represented thus:

$$SM = [-1286.06, 1398.04, -111.88] \tag{4.6}$$

$$R_{sm} = -1286.06C_1 + 1398.04C_2 - 111.88C_3 \tag{4.7}$$

$$MM = [-1238.84, 1043.6, 2541.78] \tag{4.8}$$

$$R_{mm} = -1238.84C_1 + 1043.6C_2 + 2541.78C_3 \tag{4.9}$$

The reflective coefficients are used to model the connector ORL equations represented using the symbolic math tool of Matlab and deployed in the connector ORL GUI, a sample of the code is shown on line 216 in Figure 4.4.

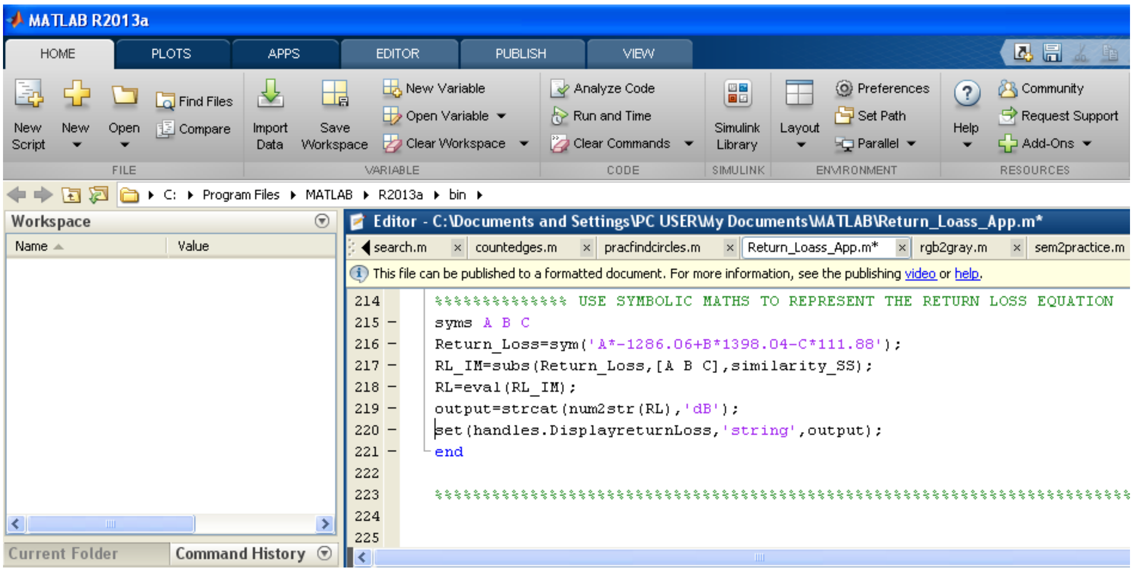


Figure 4.4: Snippet of Developed Matlab GUI Backend Symbolic Math Implementation

4.4 Connector ORL Measurement GUI.

The image processing and image analysis tools are cascaded into a single Matlab GUI which is user friendly and can be adapted to various computing platforms making a hitherto laborious process seamless and simple, a screen grab of the GUI is shown in Figure 4.5 with the source code shown in Appendix H.

The GUI comprises of a "popupmenu" to select the mode of the optical fiber, Pushbutton to browse for the enface profile image and the actual computation of the connector ORL, the central space is for the display of the binary output of the edge detected sample and the "static textbox" to display the computed connector ORL in decibels (dB).

The simulation of the model and its output are shown in Appendices I and J.

Figure 4.5: Developed Connector ORL Measurement GUI.

4.5 VALIDATION OF CONNECTOR ORL MODEL

In order to ascertain the validity of the connector ORL model incorporated in the ORL GUI (Appendix I) however, the IEC 61755-1specifies nominal ORL values for angled physical contact (APC) optical fibers as 55dB and 20dB for unmated singlemode and multimode respectively (David, 2013).

The connector ORL model results are compared with the OTDR measurements of each sample connector endface (Appendix F) and the Measurement quality jumper (MQJ) which are prefabricated calibrated fiber optic cables which accompany optical loss test sets (OLTS), the setup is shown in Figure 4.6 and the pictorial implementation is shown in Appendix F. The comparison is shown in Tables 4.2 and 4.3.

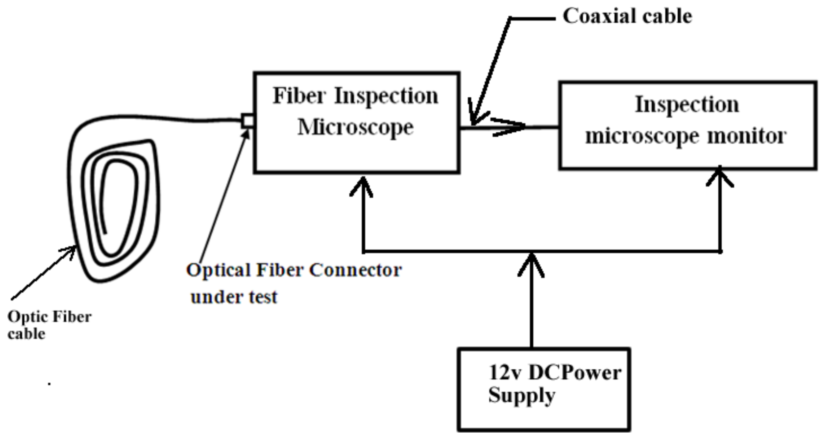


Figure 4.6: Schematic of Connector Endface Capture Setup.

Table 4.2 Multimode Connector ORL Measurements

Connector Endface Sample	**OTDR measured Return loss (dB)**	**Algorithm Computed Return loss (dB)**	**Accuracy**
Image under test 1	15.9	16.52	-0.62
Image under test 2	18.1	18.45	-0.35
Image under test 3	24.8	24.25	0.55
Reference MQJ	29.6	30.08	-0.48

Table 4.3 Singlemode Connector ORL measurements

Connector Endface Sample	**OTDR measured Return loss (dB)**	**Algorithm Computed Return loss (dB)**	**Accuracy**
Image under test 1	56.6	56.09	0.51
Image under test 2	55.7	55.32	0.38
Image under test 3	54.1.	53.45	0.65
Reference MQJ	57.6	58.24	-0.64

The developed ORL model gave a mean connector ORL accuracy of ±0.5dB and ±0.55dB for the multimode and singlemode connector endface profiles respectively as shown in equation (4.6) and (4.7).

$$\text{Multimode Mean ORL accuracy} = \frac{|-0.62|+|-0.35|+|0.55|+|-0.48|}{4} \tag{4.10}$$

$$\text{Multimode Mean ORL accuracy} = \pm 0.50\text{dB}$$

$$\text{Singlemode Mean ORL accuracy} = \frac{|0.51|+|0.38|+|0.65|+|-0.64|}{4} \tag{4.11}$$

$$\text{Singlemode Mean ORL accuracy} = \pm 0.55\text{dB}$$

The Telecommunications Industry Association (TIA) and Electronic Industries alliance (EIA) TIA/EIA-455-95-A standard also known as FOTP-95 (Absolute optical power test for optical fiber cables) specifies a measurement uncertainty of ±1dB and ±5dB for singlemode and multimode respectively (David, 2013) which is in the range of the mean accuracy of the developed connector ORL model.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The edge detection methods which are the Canny, Sobel, Prewitt, Laplacian of the Gaussian (LoG), Roberts and Frei-Chen operators were evaluated and each yielded 139.1, 49.43, 49.57, 103.2, 45.19 and 50.98 average detections per section respectively. The Canny operator was chosen as the candidate method by task-based empirical evaluation whose figure of merit is the magnitude of sum of total edge detections.

The connector ORL model was also achieved by evaluating the cross-correlation coefficient of each image region of the three (3) connector sample end faces with that of the reference MQJ resulting in three linear equations solved using the inverse matrix method using the Matlab 2013 Mfile. Equations 4.7 and 4.9 represent the developed Connector ORL model for singlemode and multimode fiber optic connectors respectively.

Furthermore the model was implemented and simulated using the Matlab 2013 GUI, symbolic Math toolbox and image processing toolbox and the results validated. The model achieved a mean connector ORL accuracy of ±0.5dB and ±0.55dB for the multimode and singlemode connectors respectively.

These accuracy levels however satisfy the TIA/EIA-455-95-A standard which specifies a measurement uncertainty of ±1dB and ±5dB for singlemode and multimode respectively. The repeatability of the results of the developed GUI was observed to be consistent after consecutive trials a feature lacking in the conventional techniques such as the Optical Reflection Discrimination (ORD) method.

5.2 Significant Contributions

The significant contributions of this research are as follows;

i. The Canny edge detection method is efficient in the processing of fiber inspection microscope connector endface profiles, which have undergone image histogram equalization because it clearly identifies the outline of the core, cladding and ferrule and yields 139.1 average detections per image section, the highest value amongst the classic operators considered.

ii. The modelling of connector ORL as a function of the discontinuities and similarity of the connector endface image profile

iii. The development of a Matlab Graphic user interface (GUI) that leverages on the ORL model in (ii) to determine connector ORL from connector endface image profile with an accuracy of ±0.5dB and ±0.55dB for multimode and singlemode respectively..

5.3 Limitation

The limitation of this dissertation is as follows:

Inability to secure an interferometer to independently verify the stated MQJ connector return loss.

5.4 Recommendation for further work

To further improve the accuracy of the model:

i. the pixel based segmentation can be implemented so as to produce more than three sub-images this will inadvertently increase the number of linear equations to be solved and may significantly increase the measurement accuracy.

ii. An inspection microscope with a magnification greater than 200X can be used to capture the connector endface image profile.

REFERENCES

Ab-Rahman, M., Ng, B., Premadi, A., and Jumari, K.(2008). A new approach for identifying Fiber fault and detecting Failure location. *International Conference on Engineering Design [ICED]* Penang, Malaysia, pp. 1-4

Ajit, C. T. and Dorothy, D. D. (2000). *Statistics and Data Analysis from elementary to Intermediate.* Prentice Hall, NJ, USA, pp. 198-201.

Annadurai, S. (2007). *Fundamentals of Digital Image Processing*. Pearson Education, India, pp. 318-370.

An-Tsung, P., Bing-Chuan, T. and Shanq-Jang, R. (2013). A Sub-Image Edge Preservation method for histogram equalization. 9th International Conference on Information, Communications and Signal Processing (ICICS), pp.10-13

Amandeep, K., Amanpreet, K., Kamaljit, S. B. and Kulwinder, S. (2013). CATV Transmission in Optical Communication System. *Journal of Optical Communications*. Vol. 34(3), pp. 209–214

Bagad, V. (2009). *Optical Fiber Communications*. Technical Publications Pune, India, pp. 2-19.

Baghai, W. (2013). 'Basic image processing: edge detection'. Retrieved from http://www.dip.ee.uct.ac.za/~nicolls/lectures/eee401f/03_edgedet_notes_2up.pdf 26/06/ 2013

Barghavi, G. (2008). *A progressive approach to Feedback-Controlled edge Detection using Boolean Derivatives*. Denmark: Proquest Inc, pp. 56-60.

Berdinskikh, T., Bragg, J., Tse, E., Daniel, J., Arrowsmith, P., Fisenko, A. & Mahmoud, S. (2002). The contamination of fiber optics connectors and their effect on optical performance. *IEEE Optical fiber Communication (OFC) Conference*. pp 617-619

Bill, W. & Emile, B. H. (2005). *Fiber Optics Installer and Technician Guide*. CA: SYBEX Alameda, pp. 2-200.

Bob, C. (2009). *Planning Fiber Optic Networks*. The McGraw-Hill Companies, Incorporated USA, pp. 45-51.

Buerli, R. (2000). Measure Return Loss in Multimode Fiber Optic Systems. *Test & Measurement World*, pp. 59-64.

Casimer, D. (2002). *Fiber Optic Data Communication: Technological Trends and Advances*. Academic Press, London, pp. 56 – 80.

Chris, S. & Toby, B. (2011). *Fundamentals of digital image processing : a practical approach with examples in Matlab*. John Wiley & Sons Ltd, Oxford, pp. 21-105,255

Corning Cable Systems Engineering Services Division (2008). "Considerations For Optical Fiber Termination", Applications Engineering Note 89, Revision 3. North Carolina, USA

David, G. (2013). *Fiber Optic Reference Guide*. Focal press, Burlington, MA, pp. 241 - 248.

Eric, S.A., Karen, P. and Sos, S. A. (2007). Quantifying image similarity using measure of enhancement by entropy. *The International Society for Optical Engineering*. Vol. 6579

Francesco, M., Antonio, C., Carlo, F., Matteo, T., Andrea, M., & Mario, M. (2010). Roughness Induced Backscattering in Optical Silicon Waveguides. *Physics Rev. Letter*, 3(104), pp. 390-392.

Fritz, A. (2011). Digital Image Analysis.Retrieved from http://www.uio.no/studier/emner/matnat/ifi/INF4300/h11/undervisningsmateriale/INF4300-2011-e04-segmentation.pdf. 0n 13/02/2015

Govind, P. A. (2002). Fiber-Optic Communications Systems (3rd Edition . John Wiley & Sons, Inc, USA, pp. 15-74, 77-116.

Haldo, S. and Juan, C. (2015). A review of Classic edge detectors. Image processing On line. Vol. 5, pp. 90 -123

Ian, T. Y. (1998). Fundamentals Of Image Processing. Retrieved from http://www.tnw.tudelft.nl/en/about-faculty/departments/imaging-physics/research/researchgroups/quantitative-imaging/courses/ on 26/04/2014, pp. 1-3

Ivan, K. (2013). Optical Fiber telecommunication volume VIA: components and subsystems. Academic Press, Massachusetts, pp. 357-362.

James, N. D. (2004). Fiber-Optic Communications. Cengage Learning, USA, pp. 6-30.

Jayaraman, S., Esakkirajan, S. &Veerakumar, T. (2011). *Digital Image Processing* . Tata McGraw-Hill Education, India, New Delhi, pp. 2-70.

John, C. R. (2011). *The Image Processing Handbook (6th Edition).* CRC Press, Florida, USA, pp 378-390.

Junzhi, G., Zongyi W. &Xiaochen, Y. (2010).A new approach for Road Centerlines

Extraction and Width Estimation. *10th IEEE International Conference on Signal Processing (ICSP) Beijing*, pp. 924-927.

Kanopoules, N.&Vasanthavada, N. and Baker, R. (2002). Design of an image edge detection filter using the Sobel Operator. *IEEE journal of Solid-State Circuits*, Vol. 23(2), pp. 358-367

Kihara, M., Morikazu, U., Masayuki, O. &Hisashi, I. (2010). Analyzing return loss deterioration of optical fiber joints with various air filled gaps over wide wavelength range. *Optical Fiber Communication collocated Fiber Optic Engineers Conference*, pp. 1-3

Kihara, M., Uchino, M., Omachi, M. & Watanabe, H. (2013). Investigation Into Optical Performance of Fiber Connections With Imperfect Physical Contact. Journal of Lightwave Technology, Vol.31 (6), pp. 967-974.

Koyama, R., Izumita, H. &Kurashima, T. (2012). Physical Contact Connection Between SC Connector and Raw Cleaved SM-Fiber End. *IEEE Photonics Technology Letters*, Vol. 24(24), pp. 2280-2283.

Kundali, H.N. (2001). The Analysis of an optical fiber communications system using laser rate equations. *Tanzania Journal of science*, 27(1), pp. 74-86

Laferrière, J., Lietaert, G., Taws, R. &Wolszczak, S. (2007). Reference Guide to Fiber Optic Testing Vol. 1. JDS Uniphase Corporation, France, pp 8-31.

Li, T. & Jean, J. (2013). *Digital Signal Processing: Fundamentals and Application*. Academic Press, Massachusetts, USA, pp.706-720

Lu, X. & Chia-Chih, C. and Aggarwal, J.K. (2011). Human detection using depth information by Kinect, *Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Computer Society Conference* , pp.15,22, 20-25.

Mai, T.V., Molnar, J. A. &Tran, L. H. (2004). Fiber optic test equipment - evaluation of OTDR dead zones and ORLM return loss. AUTOTESTCON Conference Proceedings, pp. 94-98.

Mamta, J. & Parvinder, S. S. (2009). Performance Evaluation of Edge Detection Techniques for images in spatial domain. *International Journal of Computer theory and Engineering*, 1(5), pp. 1793-8201

Maria, P. & Panagiota, B. (1999). *Image Processing: The Fundamentals*. John Wiley & Sons New Jersey, USA, pp. 125-127.

Martin, D., Fowlkes, C. & Malik.J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal.and Machine Intelligence*., 26(5), pp. 530– 549.

Mohammad S. A., Boonchuan, N., and Mohd. H. G. S., (2009). Thermal Noise Effect on Receiver Sensitivity in FTTH Customer Access Network – Simulation Approach. *Journal of Optical Communications*. Vol. 30(2), pp. 95–98

Mohammad, S. A., Boonchuan. N., Aswir, P. and Roystom, K. (2010). Low Cost Method for Testing, Troubleshooting and Monitoring of Passive Optical Network – Towards the Installation of Centralized Failure Troubleshooting System (CFTS). *Journal of Optical Communications*. Vol. 31(2), pp. 92–98

Noutsios, P. C. (2006). Optical Return Loss Measurements and Simulation of an Arbitrary Array of Concatenated Reflective Elements on Field-Installed Optical Links. *Journal Of Lightwave Technology*, Vol. 24(4), pp. 1697-1702.

Patrick, G. (2004). Foundations of Digital signal processing: Theory, Algorithms and hardware design. Institution of Engineering and Technology, UK, pp 40-50.

Peter, C. N. (2006). Optical Return Loss Measurements and Simulation of an Arbitrary Array of Concatenated Reflective Elements on Field-Installed Optical Links.*Journal OfLightwave Technology,* Vol. 24(4), pp 1697-1702

Pierre, L. (2008). *Fiber-Optic Communications*. ISTE Ltd and John Wiley & Sons, Inc, United states, pp. 90-100.
.

Qiang, L & Robert, M. N. (2015).*Computer-Aided Detection and Diagnosis in Medical Imaging*. CRC Press, Florida, pp. 98-210

Raghavender, R., Nikhil, P. and Nagabhooshanam, E. (2014). Application Of Normalized Cross Correlation to Image Registration. *International Journal of Research in Engineering and Technology*. Vol. 3(5), Pp. 12 - 16

Rangarajan, A., Chellappa, R. & Zhou, Y.T. (2002). A Model-Based Approach for Filtering and Edge Detection in Noisy Images.*IEEE Transaction on Circuits and Systems*. Vol. 37 (1), Pp. 140-144

Raman, M. &Himanshu, A. (2003). Study and Comparison of Various Image Edge Detection Techniques. *International Journal of Image Processing (IJIP),* Vol. 3(1), pp. 1-11

Rashmi, Mukesh, K., & Rohini, S. (2013). Algorithm and Technique on various edge detection: A survey. *Signal & Image Processing International Journal*, Vol.4 (3), pp 65-75.

Rongquing, H. & Maurice, O. (2009). *Fiber Optic Measurement Techniques*. Elsevier Academic Press, Massachusetts, USA, pp. 1-21, 62-67

Selim, F. (2011). Power cable inspections using MATLAB Graphical user interface aided by thermal imaging. *5th Malaysian Software Engineering Conference* , pp. 263-268.

Shikama, K., Abe, Y., Yanagi, S. & Takahashi, T. (2013). Physical-contact conditions for multicore fiber optical connectors. *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC),* pp. 17-21.

Surachet, K. (1999). Modeling and Analysis of the Effects of Impairments in Fiber Optic Links (Unpublished M.Sc. Thesis). USA: Virginia Polytechnic Institute and State University.

Tinku, A. &Ajoy, K. R. (2005). Image Processing: Principles and Applications. John Wiley & Sons, NewJersey, USA, pp. 110-111.

Tranos, Z., Oludayo, O., Sunday, O. and Seleman M. (2014). Image Segmentation, Available Techniques, Developments and Open Issues. *Canadian Journal on Image Processing and Computer Vision.* Vol. 2(3), Pp. 20 -29.

Trucco, E. and Verri, A. (1998). *Introductory techniques for 3D computer Vision.* Prentice Hall, New Jersey, USA, pp. 16-26, 68-91, 248-275

Tzafesta, S.G. (2001). *Advances in intelligent systems: Concepts, Tools and Applications*. Springer Science & Business Media, Netherlands, pp. 289-300.

Vivek, A. (2004). *Optical Network Design and Implementation*. Cisco press Indianapolis, USA, pp. 289-300.

Vijayachitra, S. (2013).*Communication Engineering*. Tata McGraw-Hill Education, India, pp. 418.

Vivien, F. & Cecilia, B. (2011).Africas ICT Infrastructure: Building on the mobile revolution. *World Bank Publications*. pp. 94-95

Wufeng, X., Lei, Z., Xuanqin, M. &Bovik, A.C. (2014). Gradient Magnitude Similarity Deviation: A Highly Efficient Perceptual Image Quality Index. IEEE *Transactions on Image Processing*, Vol.23(2), pp. 684-695.

Yajima, Y., Watanabe, H., Kihara, M. &Toyonaga, M. (2011). Optical performance of field assembly connectors using incorrectly cleaved fiber ends. *16th Opto-Electronics and Communications Conference (OECC)*, pp. 4-8.

Zuyuan, H., Mahmood, W., Sahinci, E., Ahmad, N. &Pradieu, Y. (2004). Analysis on the effects of fiber end face scratches on return loss performance of optical fiber connectors. *IEEE journal of Lightwave Technology*, 22(12), pp. 2749-2754.

APPENDICES

APPENDIX A

SOURCE CODE FOR THE IMPLEMENTATION OF EDGE DETECTION OPERATORS

```
clear all;
close all;
clc;
I = imread('C:/Documents and Settings/PC USER/My Documents/MSC 
DOCS/dirty_fiber_connector_endface.jpg');
I=rgb2gray(I);
subplot(2,2,1)
imshow(I);
title('GRAYED IMAGE VIEW')
subplot(2,2,3)
 imhist(I);
 grid on;
 title('HISTOGRAM OF GRAYED IMAGE VIEW')
 J = histeq(I);
 subplot(2,2,2)
  imshow(J);
  title('GRAYED-EQUALIZED IMAGE VIEW')
 subplot(2,2,4)
 imhist(J);
 title('Histogram of GRAYED-EQUALIZED IMAGE VIEW')
 T=multithresh(I,2)
 T1=(T(1,1);
 T2= T(1,2);
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%centroid
 %s
 %plot(centroids(:,1), centroids(:,2), 'b*')
 %hold off
 BW1 = edge(I,'canny', T2, T1);
 BW2 = edge(I,'sobel');
 BW3 = edge(I,'prewitt');
 BW4 = edge(I,'log');
 BW5 = edge(I,'roberts');
 BW6 = edge(I,'freichei’);
 bwarea(BW1); %Estimates the area of the objects in binary image
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(2)
subplot(1,2,1)
imshow(I);
title('Original Singlemode connector 
endface','FontWeight','bold','FontSize',13,...
    'FontName','Times New Roman');
%r = corr2(A,B) command for correlation coefficient.
subplot(1,2,2)
imshow(BW1);
title('CANNY OPERATOR','FontWeight','bold','FontSize',13,...
    'FontName','Times New Roman');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(3)
subplot(1,2,1)
imshow(I);
title('Original Singlemode Connector 
Endface','FontWeight','bold','FontSize',13,...
    'FontName','Times New Roman');
subplot(1,2,2)
```

```
imshow(BW3);
title('PREWITT OPERATOR','FontWeight','bold','FontSize',13,...
    'FontName','Times New Roman');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(4)
subplot(1,2,1)
imshow(I);
title('Original Singlemode connector
endface','FontWeight','bold','FontSize',13,...
    'FontName','Times New Roman');
subplot(1,2,2)
imshow(BW4);
title('LAPLACIAN of GAUSSIAN
OPERATOR','FontWeight','bold','FontSize',13,...
    'FontName','Times New Roman')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(5)
subplot(1,2,1)
imshow(I);
title('Original Singlemode connector
endface','FontWeight','bold','FontSize',13,...
    'FontName','Times New Roman')
subplot(1,2,2)
imshow(BW2);
title('SOBEL OPERATOR','FontWeight','bold','FontSize',13,...
    'FontName','Times New Roman')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(6)
subplot(1,2,1)
imshow(I);
title('Original Singlemode connector
endface','FontWeight','bold','FontSize',13,...
    'FontName','Times New Roman');
subplot(1,2,2)
imshow(BW5);
title('ROBERTS OPERATOR','FontWeight','bold','FontSize',13,...
    'FontName','Times New Roman')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(7)
subplot(1,2,1)
imshow(I);
title('Original Singlemode connector
endface','FontWeight','bold','FontSize',13,...
    'FontName','Times New Roman')
subplot(1,2,2)
imshow(BW6);
title('FREI-CHEN OPERATOR','FontWeight','bold','FontSize',13,...
    'FontName','Times New Roman')
```

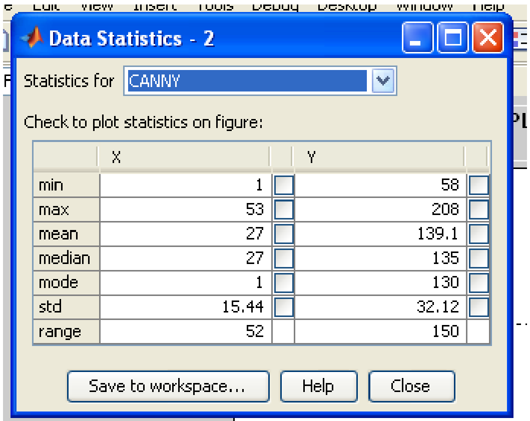
APPENDIX B

MATLAB SOURCE CODE FOR THE PLOT OF DETECTED EDGES PER SECTION OF CONNECTOR ENDFACE PROFILE

```
clear all;
close all;
clc;% some random block, simulating your case
a_blk = rand(10,5)-0.5;
% edge pixels are greater than 0 (as a assume).
sum(a_blk(:)>0);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fun = @(block_struct) sum(block_struct.data(:)>0);
%I = imread('C:/Documents and Settings/PC USER/My
Documents/MATLAB/testimage.jpg');
I = imread('C:/Documents and Settings/PC USER/My Documents/MSC
DOCS/dirty_fiber_connector_endface.jpg');
figure(3);
imshow(I);
J=rgb2gray(I);
BW1 = edge(J,'Canny');
 BW2 = edge(J,'Sobel');
 BW3 = edge(J,'Prewitt');
 BW4 = edge(J,'log');
 BW5 = edge(J,'roberts');
 BW6 = edge(J,'freichei');

%BW1 = edge(J,'Canny');
%BW2 = edge(J,'Sobel');
I1 = blockproc(BW1,[5 263],fun);
I2 = blockproc(BW2,[5 263],fun);
I3 = blockproc(BW3,[5 263],fun);
I4 = blockproc(BW4,[5 263],fun);
I5 = blockproc(BW5,[5 263],fun);
I6 = blockproc(BW6,[5 263],fun);
figure;
imshow(J);
x=linspace(1,53,53);
[m,n] = size(I2);
%subplot(1,2,1)
figure(1)
plot(x,I1(:,2),x,I2(:,2),x,I3(:,2),x,I4(:,2),x,I5(:,2),x,I6(:,2));
legend('CANNY','SOBEL','PREWITT','LoG','ROBERTS','FREI-CHEN');
figure(2)
plot(x,I1(:,1),x,I2(:,1),x,I3(:,1),x,I4(:,1),x,I5(:,1),x,I6(:,1));
legend('CANNY','SOBEL','PREWITT','LoG','ROBERTS','FREI-CHEN');
hold all;
%subplot(1,2,2)
grid on;
%figure;
%imshow(I2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%BW1 = edge(J,'Canny');
%imshow(BW1);
```

APPENDIX C

SCREENSHOT OF THE DATA STATISTICS TOOL OF PLOT OF EDGE DETECTIONS PER SECTION OF THE CONNECTOR ENDFACE PROFILE.

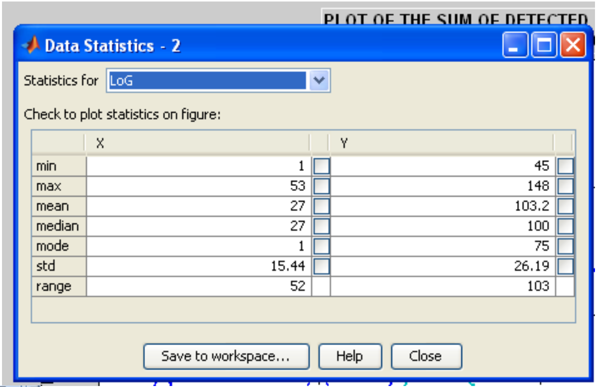

Data Statistics - 2

Statistics for CANNY

Check to plot statistics on figure:

	X	Y
min	1	58
max	53	208
mean	27	139.1
median	27	135
mode	1	130
std	15.44	32.12
range	52	150

Save to workspace... Help Close

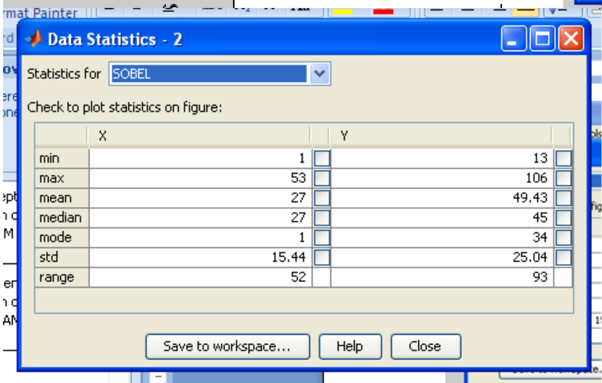

Data Statistics - 2

Statistics for LoG

Check to plot statistics on figure:

	X	Y
min	1	45
max	53	148
mean	27	103.2
median	27	100
mode	1	75
std	15.44	26.19
range	52	103

Save to workspace... Help Close

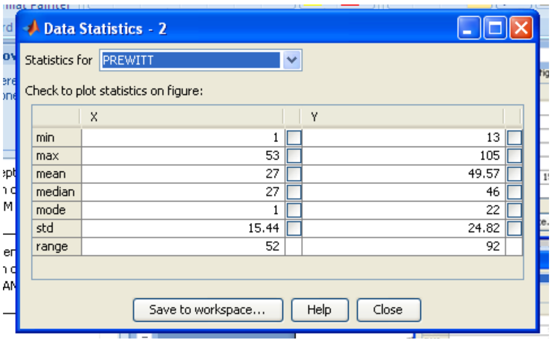

Data Statistics - 2

Statistics for SOBEL

Check to plot statistics on figure:

	X	Y
min	1	13
max	53	106
mean	27	49.43
median	27	45
mode	1	34
std	15.44	25.04
range	52	93

Save to workspace... Help Close

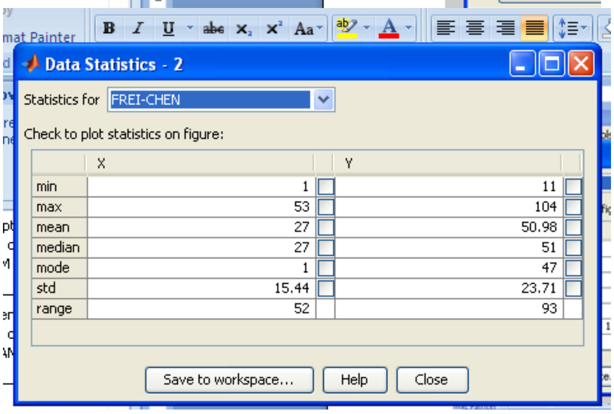

Data Statistics - 2

Statistics for PREWITT

Check to plot statistics on figure:

	X	Y
min	1	13
max	53	105
mean	27	49.57
median	27	46
mode	1	22
std	15.44	24.82
range	52	92

Save to workspace... Help Close

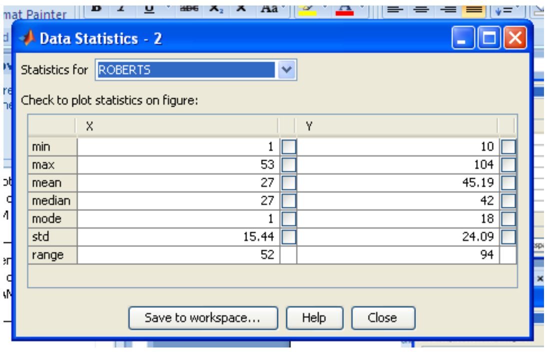

Data Statistics - 2

Statistics for FREI-CHEN

Check to plot statistics on figure:

	X	Y
min	1	11
max	53	104
mean	27	50.98
median	27	51
mode	1	47
std	15.44	23.71
range	52	93

Save to workspace... Help Close


Data Statistics - 2

Statistics for ROBERTS

Check to plot statistics on figure:

	X	Y
min	1	10
max	53	104
mean	27	45.19
median	27	42
mode	1	18
std	15.44	24.09
range	52	94

Save to workspace... Help Close

APPENDIX D

LAPLACIAN OF GAUSSIAN FILTER CODE

```
close all;
clear all;
clc;
k=3; % dIMENSION OF DESIRED kERNEL
I = imread('C:/Documents and Settings/PC USER/My Documents/MSC
DOCS/dirty_fiber_connector_endface.jpg');
J=rgb2gray(I);
sigma1=std2(J); % Standard of deviation of Kernel Desired
h1=fspecial('log',[3 3],sigma1);
```

Appendix E

Image profile sections of Canny Operator Output

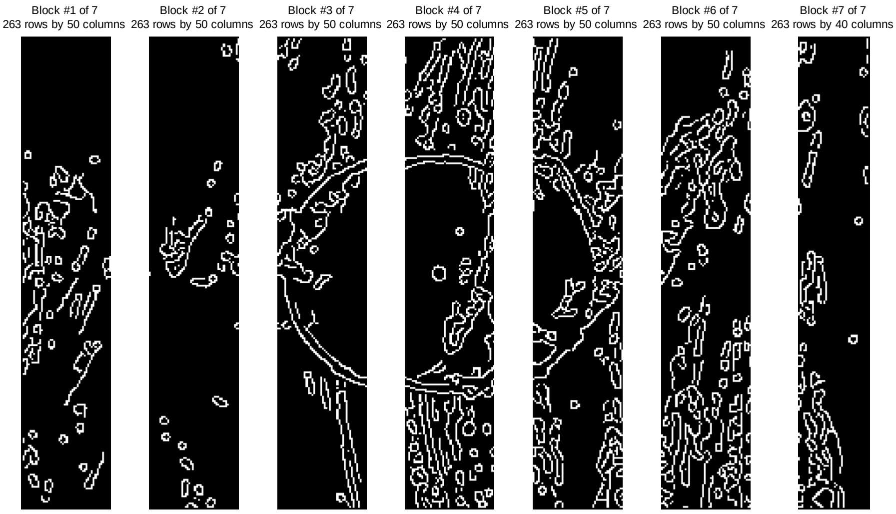
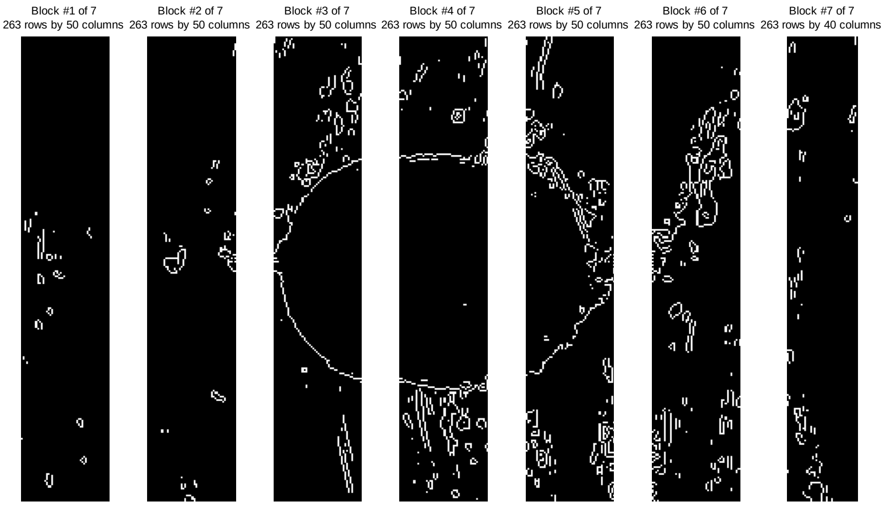


Image profile sections of Sobel operator Output



```
%MATLAB CODE FOR THE SPLITTING OF EDGE DETECTION OUTPUT IMAGE SAMPLE
%===========================================================================
======================
clear all;
close all;
clc;% some random block, simulating your caseI = imread('cameraman.tif');
I = imread('C:/Documents and Settings/PC USER/My Documents/MSC
DOCS/dirty_fiber_connector_endface.jpg');
J=rgb2gray(I);
BW1 = edge(J,'canny');
BW2 = edge(J,'sobel');
%===========================================================================
[rows,columns] = size(BW1);
% The first way to divide an image up into blocks is by using mat2cell().
blockSizeR = 263; % Rows in block.
blockSizeC = 30; % Columns in block.
% Figure out the size of each block in rows.
% Most will be blockSizeR but there may be a remainder amount of less than
that.
wholeBlockRows = floor(rows / blockSizeR);
blockVectorR = blockSizeR * ones(1, wholeBlockRows);
%blockVectorR = [blockSizeR * ones(1, wholeBlockRows), rem(rows,
blockSizeR)];
% Figure out the size of each block in columns.
wholeBlockCols = floor(columns / blockSizeC);
blockVectorC = [blockSizeC * ones(1, wholeBlockCols), rem(columns,
blockSizeC)];

% Create the cell array, ca.
% Each cell (except for the remainder cells at the end of the image)
% in the array contains a blockSizeR by blockSizeC by 3 color array.
% This line is where the image is actually divided up into blocks.

cacanny = mat2cell(BW1, blockVectorR, blockVectorC);
calog = mat2cell(BW2, blockVectorR, blockVectorC);
% Now display all the blocks.
plotIndex = 1;
numPlotsR = size(cacanny, 1);
numPlotsC = size(cacanny, 2);
for r = 1 : numPlotsR
    for c = 1 : numPlotsC
        fprintf('plotindex = %d,   c=%d, r=%d\n', plotIndex, c, r);
        % Specify the location for display of the image.
        subplot(numPlotsR, numPlotsC, plotIndex);
        % Extract the numerical array out of the cell
        % just for tutorial purposes.
        rgbBlock = cacanny{r,c};
        imshow(rgbBlock); % Could call imshow(ca{r,c}) if you wanted to.
        [rowsB, columnsB, numberOfColorBandsB] = size(rgbBlock);
        % Make the caption the block number.
        caption = sprintf('Block #%d of %d\n%d rows by %d columns', ...
            plotIndex, numPlotsR*numPlotsC, rowsB, columnsB);
        title(caption);
        drawnow;
        % Increment the subplot to the next location.
        plotIndex = plotIndex + 1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%+++++++++++++++++++++++++++++++++++++++
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Display the original image in the upper left.
```

```
figure(2);
plotIndex = 1;
for r = 1 : numPlotsR
    for c = 1 : numPlotsC
        fprintf('plotindex = %d,   c=%d, r=%d\n', plotIndex, c, r);
        % Specify the location for display of the image.
        subplot(numPlotsR, numPlotsC, plotIndex);
        % Extract the numerical array out of the cell
        % just for tutorial purposes.
        rgbBlock = calog{r,c};
        imshow(rgbBlock); % Could call imshow(ca{r,c}) if you wanted to.
        [rowsB, columnsB, numberOfColorBandsB] = size(rgbBlock);
        % Make the caption the block number.
        caption = sprintf('Block #%d of %d\n%d rows by %d columns', ...
            plotIndex, numPlotsR*numPlotsC, rowsB, columnsB);
        title(caption);
        drawnow;
        % Increment the subplot to the next location.
        plotIndex = plotIndex + 1;
    end
end
similarity=2*ones(1,numPlotsC);
for c = 1 : numPlotsC
    imseg1 = calog{1,c};
    imseg2 = cacanny{1,c};
    similarity(c)=corr2(imseg1,imseg2);
end
figure(3)
plot(similarity);
grid on;
```

Appendix F

Reference MQJ and IUT connector profile images and their OTDR measured connector return loss

Table 1: Multimode Connector endface Return loss Data

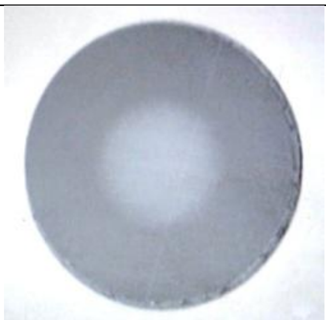
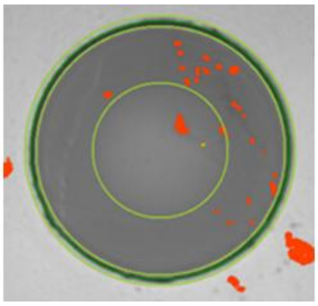
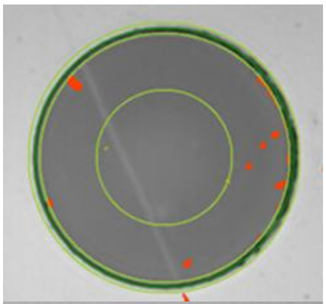
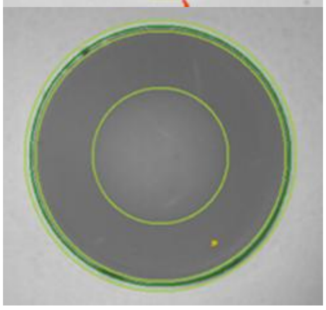
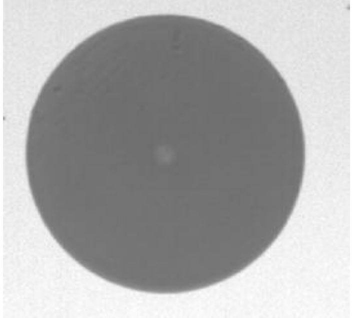
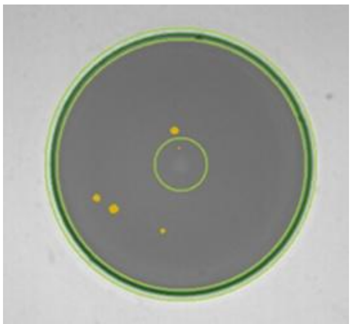
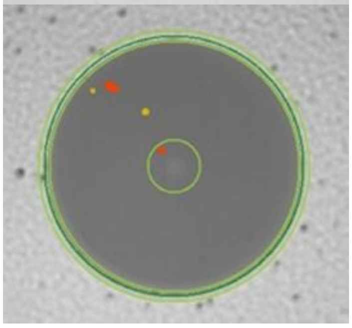
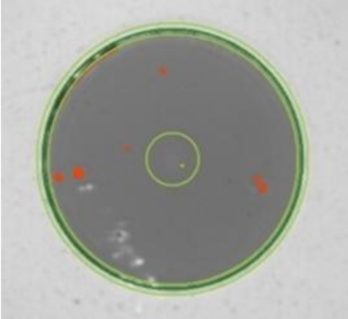
Image type	Connector Endface Profile Image	Measured OTDR Connector Return Loss (dB)
Reference Image (From Measurement quality jumper [MQJ])		29.6
Image Under Test 1		15.9
Image Under Test 2		18.1
Image Under Test 3		24.8

Table 2: Singlemode Connector endface Return Data

Image type	Connector Endface Profile Image	Measured OTDR Connector Return Loss (dB)
Reference Image (From Measurement quality jumper [MQJ])		57.6
Image Under Test 1		56.6
Image Under Test 2		55.7
Image Under Test 3		54.1

Appendix G

MalabM-file source code for connector return loss equation parameters

```
%MATLAB CODE FOR THE SPLITTING OF EDGE DETECTION OUTPUT IMAGE SAMPLE
%===========================================================================
==================
clearall;
closeall;
clc;% some random block,
Ref_Image_MM = imread('C:/Documents and Settings/PC USER/My Documents/MSC DOCS/connector
endface images/sem2ConnectorImages/SINGLEMODE REFERENCE.jpg');
Test_Image_1_MM = imread('C:/Documents and Settings/PC USER/My Documents/MSC DOCS/connector
endface images/sem2ConnectorImages/singlemode 1.jpg');
J=rgb2gray(Ref_Image_MM);
K=rgb2gray(Test_Image_1_MM);
BW1 = edge(J,'canny');
BW2 = edge(K,'canny');
%=======================================================================
[rows,columns] = size(BW1);
blockSizeR = 247; % Rows in block.
blockSizeC = 90; % Columns in block.
% Figure out the size of each block in rows.
% Most will be blockSizeR but there may be a remainder amount of less than that.
wholeBlockRows = floor(rows / blockSizeR);
blockVectorR = blockSizeR * ones(1, wholeBlockRows);
%blockVectorR = [blockSizeR * ones(1, wholeBlockRows), rem(rows, blockSizeR)];
% Figure out the size of each block in columns.
wholeBlockCols = floor(columns / blockSizeC);
blockVectorC = [blockSizeC * ones(1, wholeBlockCols), rem(columns, blockSizeC)];
cacanny = mat2cell(BW1, blockVectorR, blockVectorC);
calog = mat2cell(BW2, blockVectorR, blockVectorC);
% Now display all the blocks.
plotIndex = 1;
numPlotsR = size(cacanny, 1);
numPlotsC = size(cacanny, 2);
for r = 1 : numPlotsR
for c = 1 : numPlotsC
fprintf('plotindex = %d,   c=%d, r=%d\n', plotIndex, c, r);
% Specify the location for display of the image.
subplot(numPlotsR, numPlotsC, plotIndex);
% Extract the numerical array out of the cell
% just for tutorial purposes.
rgbBlock = cacanny{r,c};
imshow(rgbBlock); % Could call imshow(ca{r,c}) if you wanted to.
    [rowsB, columnsB, numberOfColorBandsB] = size(rgbBlock);
% Make the caption the block number.
caption = sprintf('Block #%d of %d\n%d rows by %d columns', ...
plotIndex, numPlotsR*numPlotsC, rowsB, columnsB);
title(caption);
drawnow;
% Increment the subplot to the next location.
plotIndex = plotIndex + 1;
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%+++++++++++++++++++++++++++++
++++++++++
% Display the original image in the upper left.
figure(2);
```

```
plotIndex = 1;
for r = 1 : numPlotsR
for c = 1 : numPlotsC
fprintf('plotindex = %d,   c=%d, r=%d\n', plotIndex, c, r);
% Specify the location for display of the image.
subplot(numPlotsR, numPlotsC, plotIndex);
% Extract the numerical array out of the cell
% just for tutorial purposes.
rgbBlock = calog{r,c};
imshow(rgbBlock); % Could call imshow(ca{r,c}) if you wanted to.
        [rowsB, columnsB, numberOfColorBandsB] = size(rgbBlock);
% Make the caption the block number.
caption = sprintf('Block #%d of %d\n%d rows by %d columns', ...
plotIndex, numPlotsR*numPlotsC, rowsB, columnsB);
title(caption);
drawnow;
% Increment the subplot to the next location.
plotIndex = plotIndex + 1;
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%MPUTE THE CORRELATION BETWEEN THE REF AND TEST IMAGE PROFILE
similarity=2*ones(1,numPlotsC);
for c = 1 : numPlotsC
    imseg1 = calog{1,c};
    imseg2 = cacanny{1,c};
similarity(c)=corr2(imseg1,imseg2);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(3)
plot(similarity);
gridon;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% USE SYMBOLIC MATHS TO REPRESENT THE RETURN LOSS EQUATION
```

Appendix H

Malab Graphic User Interface [GUI] Source Code

```
functionvarargout = Return_Loass_App(varargin)
% RETURN_LOASS_APP MATLAB code for Return_Loass_App.fig
%      RETURN_LOASS_APP, by itself, creates a new RETURN_LOASS_APP or raises the existing
%      singleton*.
%
%      H = RETURN_LOASS_APP returns the handle to a new RETURN_LOASS_APP or the handle to
%      the existing singleton*.
%
%      RETURN_LOASS_APP('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in RETURN_LOASS_APP.M with the given input arguments.
%
%      RETURN_LOASS_APP('Property','Value',...) creates a new RETURN_LOASS_APP or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before Return_Loass_App_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to Return_Loass_App_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 Return_Loass_App

% Last Modified by GUIDE v2.5 02-Apr-2015 10:26:43

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

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


% --- Executes just before Return_Loass_App is made visible.
functionReturn_Loass_App_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 Return_Loass_App (see VARARGIN)
```

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

% Update handles structure
guidata(hObject, handles);


% UIWAIT makes Return_Loass_App wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
functionvarargout = Return_Loass_App_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 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)

 % --- Executes on selection change in fiber_mode.
functionfiber_mode_Callback(hObject, eventdata, handles)
% hObject    handle to fiber_mode (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns fiber_mode contents as cell array
%        contents{get(hObject,'Value')} returns selected item from fiber_mode

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

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

% --- Executes on button press in ComputeReturnLoss.
functionComputeReturnLoss_Callback(hObject, eventdata, handles)
% hObject    handle to ComputeReturnLoss (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%Determine the selected data set.
I=get(handles.edit2,'String');
str = get(handles.fiber_mode, 'String');
val = get(handles.fiber_mode,'Value');
```

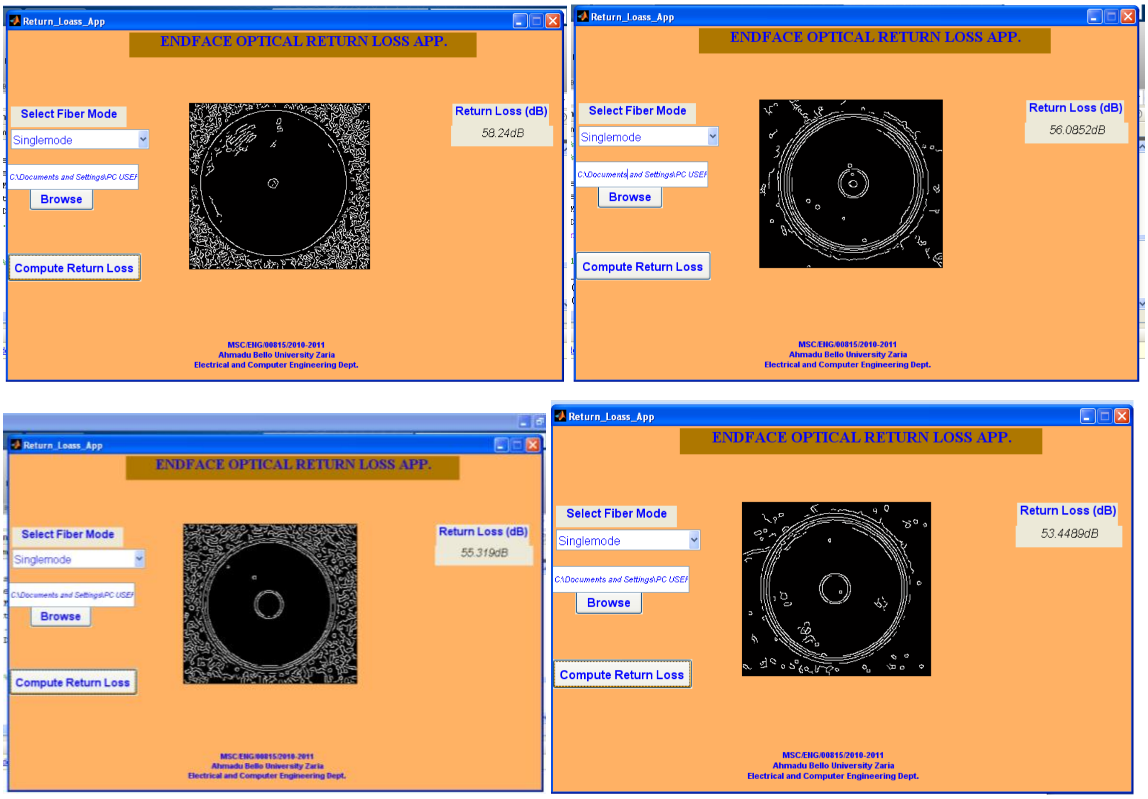
```
% Set current data to the selected data set.
switchstr{val};
case'Multimode'% User selects SINGLEMODE.
Ref_Image_MM = imread('C:/Documents and Settings/PC USER/My Documents/MSC DOCS/connector endface images/sem2ConnectorImages/MULTIMODE REFERENCE.jpg');
%Test_Image_1_MM = imread('C:/Documents and Settings/PC USER/My Documents/MSC DOCS/connector endface images/sem2ConnectorImages/multimode 2 58dB Fail.jpg');
Test_Image_1_MM=imread(I);
J=rgb2gray(Ref_Image_MM);
K=rgb2gray(Test_Image_1_MM);
BW1 = edge(J,'canny');
BW2 = edge(K,'canny');
%==================================================================
[rows,columns] = size(BW1);
% The first way to divide an image up into blocks is by using mat2cell().
blockSizeR = 231; % Rows in block.
blockSizeC = 90; % Columns in block.
% Figure out the size of each block in rows.
% Most will be blockSizeR but there may be a remainder amount of less than that.
wholeBlockRows = floor(rows / blockSizeR);
blockVectorR = blockSizeR * ones(1, wholeBlockRows);
%blockVectorR = [blockSizeR * ones(1, wholeBlockRows), rem(rows, blockSizeR)];
% Figure out the size of each block in columns.
wholeBlockCols = floor(columns / blockSizeC);
blockVectorC = [blockSizeC * ones(1, wholeBlockCols), rem(columns, blockSizeC)];

% Create the cell array, ca.
% Each cell (except for the remainder cells at the end of the image)
% in the array contains a blockSizeR by blockSizeC by 3 color array.
% This line is where the image is actually divided up into blocks.

cacanny = mat2cell(BW1, blockVectorR, blockVectorC);
calog = mat2cell(BW2, blockVectorR, blockVectorC);
% Now display all the blocks.
numPlotsC = size(calog, 2);
axes(handles.ConnectorEndfaceOutput);
imshow(BW2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%COMPUTE THE CORRELATION BETWEEN THE REF AND TEST IMAGE PROFILE
similarity=2*ones(1,numPlotsC);
for c = 1 : numPlotsC
   imseg1 = calog{1,c};
   imseg2 = cacanny{1,c};
similarity(c)=corr2(imseg1,imseg2);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%% USE SYMBOLIC MATHS TO REPRESENT THE RETURN LOSS EQUATION
symsABC
Return_Loss=sym('A*-1238.84+B*-1043.6+C*2541.78');
RL_IM=subs(Return_Loss,[A B C],similarity);
RL=eval(RL_IM);
set(handles.DisplayreturnLoss,'string',num2str(RL));
case'Singlemode'
Ref_Image_SS = imread('C:/Documents and Settings/PC USER/My Documents/MSC DOCS/connector endface images/sem2ConnectorImages/SINGLEMODE REFERENCE.jpg');
```

```
%Test_Image_1_MM = imread('C:/Documents and Settings/PC USER/My Documents/MSC DOCS/connector endface images/sem2ConnectorImages/multimode 2 58dB Fail.jpg');
Test_Image_1_SS=imread(I);
J_S=rgb2gray(Ref_Image_SS);
K_S=rgb2gray(Test_Image_1_SS);
BW1_S = edge(J_S,'canny');
BW2_S = edge(K_S,'canny');
%=========================================================================
[rows_S,columns_S] = size(BW1_S);
% The first way to divide an image up into blocks is by using mat2cell().
blockSizeR_S = 247; % Rows in block.
blockSizeC_S = 90; % Columns in block.
% Figure out the size of each block in rows.
% Most will be blockSizeR but there may be a remainder amount of less than that.
wholeBlockRows_S = floor(rows_S / blockSizeR_S);
blockVectorR_S = blockSizeR_S * ones(1, wholeBlockRows_S);
%blockVectorR = [blockSizeR * ones(1, wholeBlockRows), rem(rows, blockSizeR)];
% Figure out the size of each block in columns.
wholeBlockCols_S = floor(columns_S / blockSizeC_S);
blockVectorC_S = [blockSizeC_S * ones(1, wholeBlockCols_S), rem(columns_S, blockSizeC_S)];

% Create the cell array, ca.
% Each cell (except for the remainder cells at the end of the image)
% in the array contains a blockSizeR by blockSizeC by 3 color array.
% This line is where the image is actually divided up into blocks.

cacanny = mat2cell(BW1_S, blockVectorR_S, blockVectorC_S);
ca_TEST = mat2cell(BW2_S, blockVectorR_S, blockVectorC_S);
% Now display all the blocks.
numPlotsC_S = size(ca_TEST, 2);
axes(handles.ConnectorEndfaceOutput);
imshow(BW2_S);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%COMPUTE THE CORRELATION BETWEEN THE REF AND TEST IMAGE PROFILE
similarity_SS=2*ones(1,numPlotsC_S);
for c = 1 : numPlotsC_S
    imseg1_SS = ca_TEST{1,c};
    imseg2_SS = cacanny{1,c};
similarity_SS(c)=corr2(imseg1_SS,imseg2_SS);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%USE SYMBOLIC MATHS TO REPRESENT THE RETURN LOSS EQUATION
symsABC;
Return_Loss=sym('A*-1286.06+B*1398.04-C*111.88');
RL_IM=subs(Return_Loss,[A B C],similarity_SS);
RL=eval(RL_IM);
output=strcat(num2str(RL),'dB');
set(handles.DisplayreturnLoss,'string',output);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

functionDisplayreturnLoss_Callback(hObject, eventdata, handles)
% hObject    handle to DisplayreturnLoss (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 DisplayreturnLoss as text
%        str2double(get(hObject,'String')) returns contents of DisplayreturnLoss as a double


% --- Executes during object creation, after setting all properties.
functionDisplayreturnLoss_CreateFcn(hObject, eventdata, handles)
% hObject    handle to DisplayreturnLoss (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.
ifispc&&isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end


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.
ifispc&&isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end


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

[filename, pathname] = uigetfile({'*.jpg';'*.tif';'*.png';'*.bmp';'*.*'},'Select Connector Endface Image File');
% Now you can set the "string" property of the static textbox.
set(handles.edit2,'String',fullfile(pathname, filename));
% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over BrowseFileButton.
functionBrowseFileButton_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to BrowseFileButton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

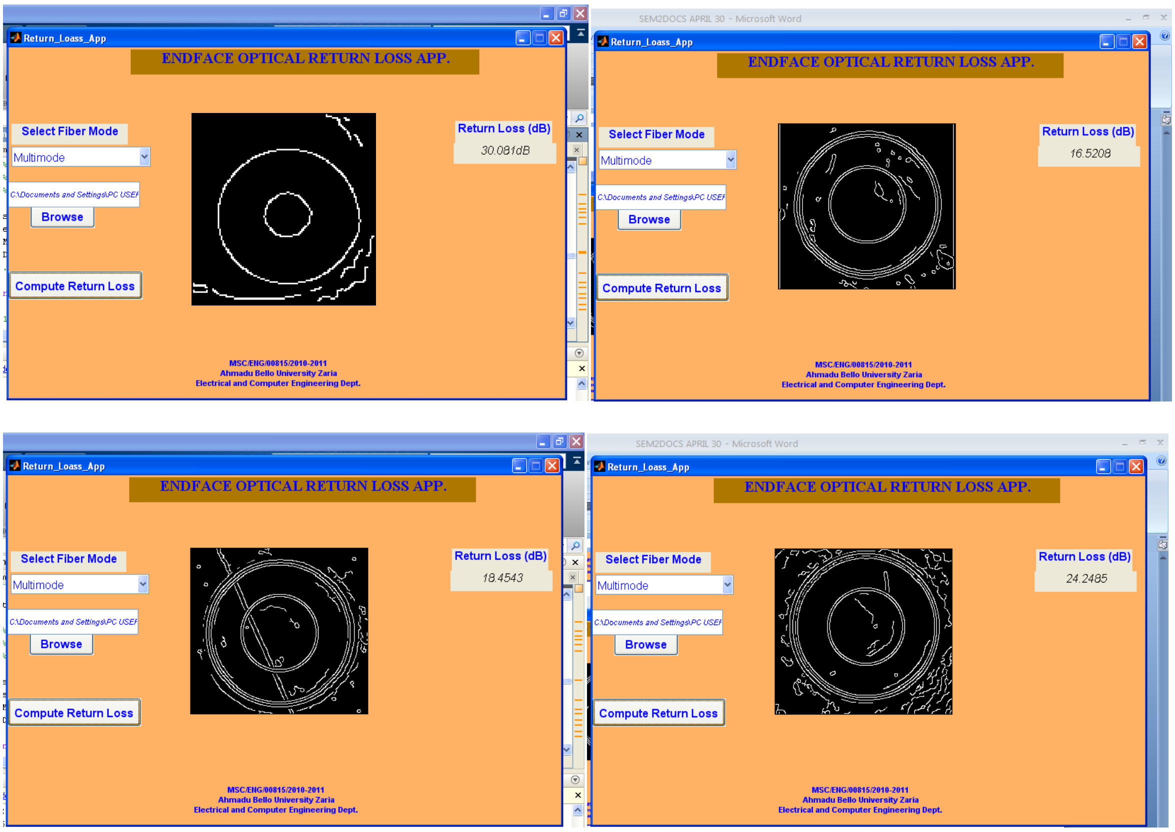
Appendix I

Singlemode Optical Return Loss Measurement GUI Interface.



Appendix J

Multimode Optical Return Loss Measurement GUI Interface.



Appendix K

Setup of OTDR and Inspection microscope measurements.

Appendix L

TYPICAL OPTICAL FIBER CABLE MANUFACTURERES SPECIFICATIONS

Optical Specifications

Fiber Attenuation

Maximum Attenuation

Wavelength (nm)	Maximum Value* (dB/km)
1310	0.33 – 0.35
1383**	0.31 – 0.35
1550	0.19 – 0.20
1625	0.20 – 0.23

*Maximum specified attenuation value available within the stated ranges.
**Attenuation values at this wavelength represent post-hydrogen aging performance.
Alternate attenuation offerings available upon request.

Attenuation vs. Wavelength

Range (nm)	Ref. λ (nm)	Max. α Difference (dB/km)
1285 – 1330	1310	0.03
1525 – 1575	1550	0.02

The attenuation in a given wavelength range does not exceed the attenuation of the reference wavelength (λ) by more than the value α.

Macrobend Loss

Mandrel Diameter (mm)	Number of Turns	Wavelength (nm)	Induced Attenuation* (dB)
32	1	1550	≤0.05
50	100	1310	≤0.05
50	100	1550	≤0.05
60	100	1625	≤0.05

*The induced attenuation due to fiber wrapped around a mandrel of a specified diameter.

Point Discontinuity

Wavelength (nm)	Point Discontinuity (dB)
1310	≤0.05
1550	≤0.05

Cable Cutoff Wavelength (λ_{ccf})

$\lambda_{ccf} \le 1260$ nm

Mode-Field Diameter

Wavelength (nm)	MFD (μm)
1310	9.2 ± 0.4
1550	10.4 ± 0.5

Dispersion

Wavelength (nm)	Dispersion Value [ps/(nm•km)]
1550	≤18.0
1625	≤22.0

Zero Dispersion Wavelength (λ_0): 1302 nm $\le \lambda_0 \le$ 1322 nm
Zero Dispersion Slope (S_0): ≤ 0.089 ps/(nm²•km)

Polarization Mode Dispersion (PMD)

	Value (ps/√km)
PMD Link Design Value	≤0.06*
Maximum Individual Fiber	≤0.2

*Complies with IEC 60794-3: 2001, Section 5.5, Method 1, (m = 20, Q = 0.01%), September 2001.

The PMD link design value is a term used to describe the PMD of concatenated lengths of fiber (also known as PMD_Q). This value represents a statistical upper limit for total link PMD. Individual PMD values may change when fiber is cabled. Corning's fiber specification supports network design requirements for a 0.20 ps/√km maximum PMD.

Dimensional Specifications

Glass Geometry

Fiber Curl	≥ 4.0 m radius of curvature
Cladding Diameter	125.0 ± 0.7 μm
Core-Clad Concentricity	≤ 0.5 μm
Cladding Non-Circularity	≤ 0.7%

Coating Geometry

Coating Diameter	245 ± 5 μm
Coating-Cladding Concentricity	<12 μm

Environmental Specifications

Environmental Test	Test Condition	Induced Attenuation 1310 nm, 1550 nm & 1625 nm (dB/km)
Temperature Dependence	-60°C to +85°C*	≤0.05
Temperature Humidity Cycling	-10°C to +85°C* up to 98% RH	≤0.05
Water Immersion	23°± 2°C	≤0.05
Heat Aging	85°± 2°C*	≤0.05
Damp Heat	85°C at 85% RH	≤0.05

*Reference temperature = +23°C