Color Medical Image Edge Detection based on Higher Dimensional Fourier Transforms Applied in Diabetic Retinopathy Studies

By:

Abebe Belachew Mekonnen

A thesis submitted in partial fulfillment of the requirements for the Degree of Master of Science in Biomedical Engineering

Center of Biomedical Engineering

Addis Ababa Institute of Technology

Addis Ababa University

Advisor: Dawit Assefa Haile (PhD)

Addis Ababa, Ethiopia, April 25, 2019

Declaration

I, the undersigned, declare that this thesis is my original work. It has never been presented for a degree in any other institution and that all sources of materials used in it have been duly acknowledged.

Name: _____

Signature: _____

Date: _____

This MSC thesis has been submitted for examination with my approval as an advisor.

_____

Dawit Assefa Haile (PhD)

Addis Ababa University

School of Graduate Studies

Certificate of Examination

This is to certify that the thesis prepared by Abebe Belachew Mekonnen entitled: *Color Medical Image Edge Detection based on Higher Dimensional Fourier Transforms Applied in Diabetic Retinopathy Studies* submitted in partial fulfillment of the requirements for the degree of Master of Science in Biomedical Engineering (Bioinstrumentation and Imaging stream) complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the examining committee

Examiner _____Signature _____ Date _____

Examiner _____Signature _____ Date _____

Examiner _____Signature _____ Date _____

Advisor _____Signature _____ Date _____

_____

Chief of Department or Graduate Program Coordinator.

Abstract

Various edge detection techniques for color images that have been proposed in the last two decades showed that color images contain 10% additional edge information as compared to their gray scale counterparts. For color image edge detection, the traditional methods used for gray-scale images are usually extended and applied to the three-color channels separately. This leads to lose the intrinsic inter-correlation information embedded in color image components in addition to computational complexity incurred. Efficient and accurate edge detection leads to increased performance of subsequent image processing and analysis techniques including image segmentation and quantification.

In this thesis, an edge detection algorithm has been proposed that treats color value triplets as vectors based on higher dimensional algebra. A human perception-based color space has been used due to its importance in color image edge detection. The trinion based algorithm has provided an efficient method to represent the color information vectorally. Color edge features are extracted based on a second order statistical technique using the weibull distribution method. A suitable color space transformation and a way of extracting robust higher order features are included in the method. Performance of the proposed scheme is compared against classical and other vectorial approaches which have been proposed in the literature based on objective criteria. Results showed that the proposed approach wins the other techniques available in the literature. Application of the proposed method has been shown in edge detection applied on color images taken from patients treated for diabetic retinopathy acquired from publicly available databases and St. Paul's Hospital Millennium Medical College. The algorithm performs well in detecting exudates, hemorrhages, optical disc and blood vessels.

Keywords - **Color image analysis, Trinion, Quaternion, Opponent color space, Localized window, Central voxel, Diabetic Retinopathy, Hemorrhage, Exudates, Color gradient, Weibull distribution.**

Acknowledgement

I would first like to thank my thesis advisor Dr. Dawit Assefa Haile (PhD) from the Center of Biomedical Engineering, AAiT, AAU. Dr Dawit`s office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently provided me with continuous support and encouragement throughout my study and process of researching. He had been following and steered me in the right direction. This accomplishment would not have been possible without him.

I am also grateful to Dr. Masreshaw Demelash (PhD), head of the Center of Biomedical Engineering, AAiT, AAU for all his help, positive attitude and encouragement to get the right data for my research. Special thanks to St. Paul's Hospital Millennium Medical College Ophthalmology Department, Dr Girum W/Gebriel, Dr Habtom and Mr Dereje Temesgen to mention a few in supporting to get the data.

Finally, I would like to thank my family and friends who have been following me to strive towards my goal.

Table of Contents

List of Tables

List of Figures

Acronyms

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
4D	Four Dimensional
CCD	Charged Coupled Device
CIE	Commission Internationale de L'Éclairage
CMYK	Cyan Magneta Yellow Black
CO	Conditional Ordering
DRS	Digital Retinopathy System
DFT	Discrete Fourier Transform
DoG	Difference of Gaussian
ITFT	Inverse Trinion Fourier Transform
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FOM	Figure of Merit
FR	Fourier Transform
HIS	Hue Intensity Saturation
HPC	High performance computer
HSL	Hue Saturation Luminousity
HSV	Hue Saturation Value
HVS	Human Visual System
IIR	Infinite Impulse Response
LoG	Laplacian of Gaussian
LPF	Low Pass Filter
MO	Marginal Ordering

MRI	Magnetic Resonance Imaging
MVD	Minimum Vector Dispersion
MVR	Minimum Vector Range
OCT	Optical Coherence Tomography
OD	Optic Disc
PO	Partial Ordering
QFT	Quaternion Fourier Transform
RGB	Red Green Blue
RO	Reduced Ordering
SPD	Spectral Power Distribution
TFT	Trinion Fourier Transform
VD	Vector Dispersion
VOS	Vector Order Statistics
VR	Vector Range

Chapter One

1. Introduction

1.1. Bckground

With the advent and development of modern electronic technologies in recent years, more and more devices are digitized and hence need built in systems. Electronic devices use their built-in sensors to capture the image signals. However, image signals are not always clean and may be accompanied with sensing noises, not directly used for further interpretation and analysis in general. In order to obtain useful features, it needs digital image processing and analysis skills [1, 2]. Bedsides, diagnostic imaging devices become a vital tool in medicine today. These imaging techniques provide an effective means for noninvasive mapping of the anatomy of an organ of a patient. These technologies have greatly increased knowledge of normal and diseased anatomy for medical research and are a critical component in diagnosis and treatment planning. With the increasing size and number of diagnostic medical images, the use of computers for processing and analysis of these medical images becomes necessary and critical. The relative change in size, shape and the spatial relationships between anatomical structures obtained from intensity, color or textural distribution provide important information in clinical diagnosis for monitoring and detection of disease progression [3].

Particularly, radiologists are interested to observe the size, shape and texture of the organs and/or parts of the organ to identify the problems. The recognition, labeling and the quantitative measurement of specific objects and structures are involved in the analysis of medical images. To provide automatic information about an object clinically in terms of its morphology and anatomy, image segmentation and classification are important tools to obtain the desired information [4]. Here edge detection is fundamental to implement image segmentation and classification successfully. Inaccuracies in edge detection directly influence the results of a subsequent feature-based image processing technique, such as region segmentation, stereo analysis, data coding, image retrieval, data hiding, watermarking, or recognition and tracking of objects in image sequences [5].

Edge detection is primary and important task in image processing and computer vision because of its different applications [6, 7] as seen in Figure 1.1. Even though a great number of edge detection approaches have been proposed in literature so far, there is still a continuing research effort to improve further. Recently, the main interest has been directed towards algorithms applied to color and multispectral images [1, 6, 8]. Compared to the grey level images, the color and multispectral images contain richer information [6]. The advantage of color edge detection schemes over grey-scale approaches is easily demonstrated by considering the fact that those edges that exist at the boundary between regions of different colors cannot be detected in grey-scale techniques if there is no change in gray scale intensity (see Figure 1.2, for example). In many situations, strong edges between colors become very weak in luminance and are difficult to detect.

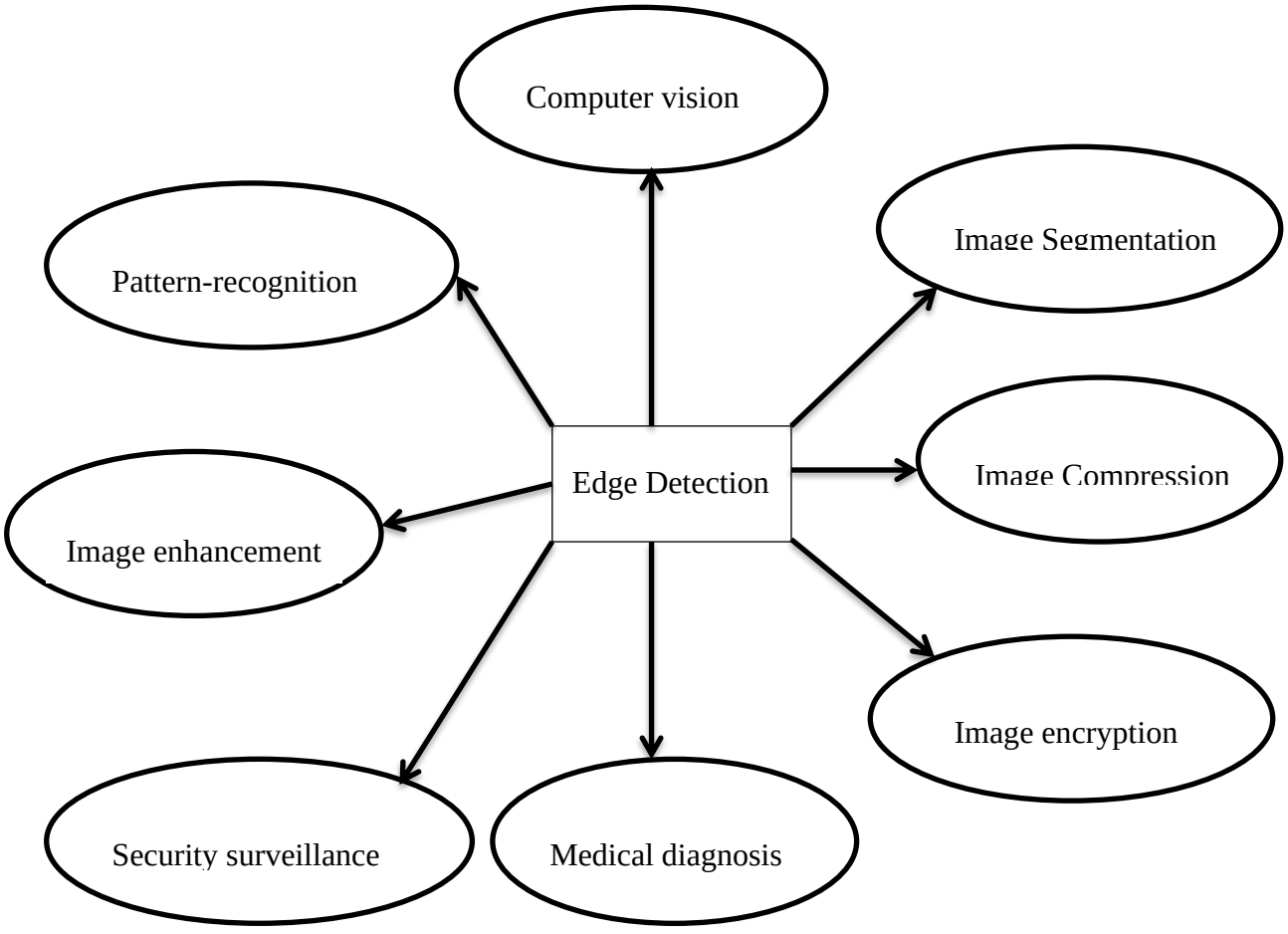


Figure 1.1 Applications of edge detection system [7].

Studies showed that 10% edges would be lost by transforming a color image to its grey-scale counterpart. To demonstrate this concept, a Matlab generated synthetic color image which loses the important information by transforming to its gray scale counterpart is shown in Figure 1.2. Here, it is natural to understand that none of the classical edge detection techniques can detect the edges of the synthetic Matlab generated color image shown in the figure since they all rely on gray scale intensity information. We can see that the vector valued (trinion based) technique is in a good stand to detect edges in places with no intensity differences.

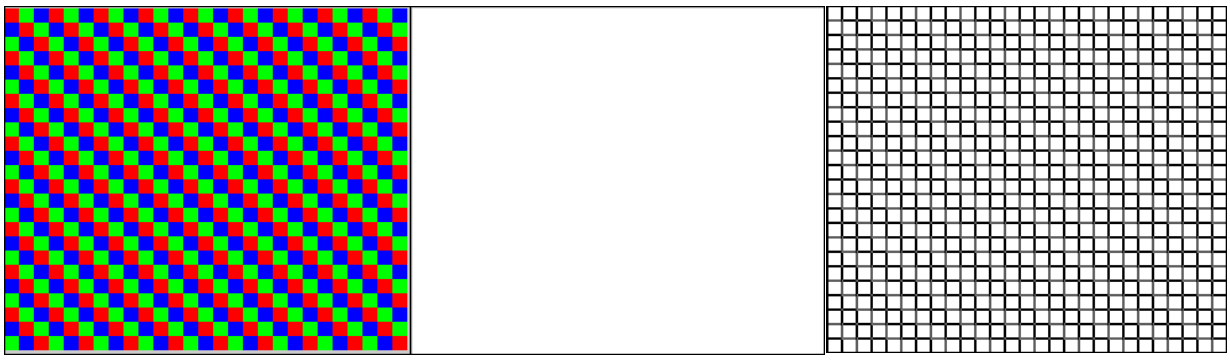


Figure 1.2 Matlab generated synthetic color image consisting of three different color squares of similar intensity in a grid pattern (left), its corresponding gray scale image (middle) and results of vector-valued (Trinion based) edge (right).

The early monochromatic methods for color edge detection calculate the gradient of a single component image by using first derivative (gradient) or second derivative operators. Roberts, Prewitt, Sobel, Laplacian, Canny, and others are amongst those methods [6, 9]. These methods are computationally simple, but ignore the correlation information among the Red, Green and Blue components in Red-Green-Blue (RGB) color space [10, 11, 12]. Some edges caused by hue changes would be missed. Based on comparison of different color image edge detectors in multiple color spaces, Wesolkowski et al. [6] concluded that it was more important to find the appropriate color space representation when one wants to design a robust color edge detection algorithm.

Edge recognition is the name for a set of mathematical methods which target at classifying points in an image at which the image intensity varies sharply or has discontinuities. The points at which digital image intensity turns sharply are stereotypically ordered into a set of line segments called edges. Edge detection is a common image processing task that often forms the initial stage

of an automated image interpretation [13, 14, 15]. It is an essential instrument in vision processing, machine vision and digital image processing, mainly in the areas of feature recognition [16]. However, edge detection could be a very subjective task. Two or more observers could work on the same image and may end up identifying edges which may not agree with each other [16]. As a user of an edge detector, one should not imagine the software to automatically detect all the edges he/she desires and nothing more, because a program cannot possibly know what level of details the experimenter has in mind [16].

As edge detection is commonly used operation in image analysis and there are probably more algorithms in the literature for enhancing and detecting edges than any other single subject. The reason for this is that edges form the outline of an object. An edge is the boundary between an object and the background which indicates the boundary between overlapping objects. This means that if the edges in an image can be identified accurately, all the objects can be located and basic properties such as area, perimeter and shape can be measured to be used for different applications [17].

There is a fundamental difference between color images and gray scale images. In a color image, a color vector (which generally consists of three components) is assigned to a pixel, while a scalar gray scale is assigned to a pixel of a gray scale image. Thus, in color image processing, vector valued image functions are treated instead of scalar image functions as in gray scale image processing. Accordingly, two basic classes of color image processing approaches exist in the literature [1, 15, 18]:

- Monochromatic based (serial) techniques: These techniques treat information from the individual color channels or color vector components separately and then combine the individual results gained [1, 15, 18].
- Vector valued techniques: These techniques treat the color information as color vectors in a vector space and treated as one entity or object [1, 15, 18].

1.2. Statement of the Problem

Techniques which are classified as modern still treat color images as separated monochromatic images while detecting edges. But such methods have a potential drawback because by taking the different bands separately, we will lose the important information about how the different

color components are correlated to each other. Also, since we are dealing with each component separately, the computation involved to extract edges will be high. The right solution is to develop a method that incorporates all the color components as one entity and keep the correlations between them intact. In this regard, there are more holistic approaches proposed in the literature to represent and analyze colors. It is the intent of this thesis to investigate the application of such methods for edge detection of color medical images and develop a robust algorithm that could circumvent some of the drawbacks of other edge detection tools commonly used in the literature.

1.3. Objectives

1.3.1. General Objective

To develop a holistic and efficient edge detection algorithm for use in color medical images.

1.3.2. Specific Objectives

- To investigate applications of holistic approaches of color image representation and analysis in edge detection of medical images;
- Propose a robust color image edge detection algorithm;
- Test the efficacy of the proposed algorithm on standard test images;
- Show the application of the proposed algorithm in edge detection of retinal color images generated from patients diagnosed with Diabetic Retinopathy;
- Develop quantitative evaluation criteria to evaluate the performance of the proposed algorithm and perform comparison against other color edge detection tools already available in the literature;

1.4. Significance of the Study

Many image processing applications including image segmentation, classification and object recognition could benefit from a clever edge detection tool. Image based delineation of tumors and other lesions, for example, highly rely on accurate identification and localization of their boundaries/edges. When such procedures are done manually, it is often time consuming, subjective and hardly repetitive. Automating the procedure hence has vital importance. The developed automated edge detection scheme in the current study is meant to circumvent the above-mentioned issues to a considerable extent. Developing a robust and computationally

efficient color edge detection tool overcomes some of the limitations of edge detection tools already proposed in the literature. Application of the proposed tool has been shown on images generated from patients treated for Diabetic Retinopathy (DR), which is a known complication of diabetics. Detection and localization of DR features, including hemorrhages, hard and soft exudates and Optical Disc (OD) has paramount importance for both objective mass screening as well as clinical settings. Furthermore, the approach presented in this thesis could potentially be applied in cases of non-medical color images generated in other disciplines showing its great promises.

1.5. Scope and Limitations of the Study

This research was conducted to develop holistic and robust color edge detection algorithm for use in analyzing color medical images. Validation of the proposed color edge detection tool has been shown by evaluating its performance when applied on standard test color images. The performance of the proposed algorithm has also been tested on images acquired from St. Paul's Hospital Millennium Medical College as well as two other online databases. The algorithm has been tested in extracting important features like exudates, hemorrhages, blood vessels, OD and other structures from DR images to be used for further image analysis techniques. Furthermore, the research implementation of the algorithm has been shown entirely on a Matlab platform.

1.6. Organization of the Thesis

The rest of the thesis has been organized into five chapters. Chapter **two** presents literature review on different edge detection methods including both classical and vectorial approaches. **Chapter three** discusses the basic concepts of medical image processing and available color spaces as well as the mathematics behind color space transformation. **Chapter four** presents the proposed color medical image edge detection algorithm, selection of an optimal color space (opponent color space), a holistic representation and analysis of the resulting color images, and ways of extracting meaningful higher order statistical features for use in generating edge maps. **Chapter five** presents data set that have been used in the current study, result and discussion points on some performance measures while **chapter six** presents concluding remarks and possible future directions of the study.

References

[1]. Yu-Zhe Hsiao and Soo-Chang Pei, “Edge detection, color quantization, segmentation, texture removal, and noise reduction of color image using quaternion iterative filtering”, Journal of Electronic Imaging 23(4), 043001, Jul/Aug 2014.

[2]. Phairoj Samutrak and Jeeraporn Werapun, “Efficient Color Edge Detection based on Synthetic Weighted multi-Structure Element Morphology”, Fifth International Conference on Graphic and Image Processing (ICGIP 2013), Proc. of SPIE, 9069(1), Dec 2013, doi: 10.1117/12.2051035.

[3]. Indra Kanta Maitra, Sanjay Nag and Samir K. Bandyopadhyay, “A Novel Edge Detection Algorithm for Digital Mammogram”, International Journal of Information and Communication Technology Research, 2(2), February 2012.

[4]. Todd A. Ell, Nicolas Le Bihan and Stephen J. Sangwine, Quaternion Fourier Transforms for Signal and Image Processing, Digital Signal and Image Processing Series, John Wiley & Sons, Inc., 2014.

[5]. A. Koschan and M. Abidi, “Detection and Classification of Edges in Color Images”, Signal Processing Magazine, Special Issue on Color Image Processing, 22(1), pp. 64-73, 2005.

[6]. Tao Lei, Yangyu Fan, Yi Wang, “Color edge detection based on the fusion of hue component and principal component analysis”, IET Image Process., 8(1), pp. 44–55, 2014, doi: 10.1049/iet-ipr.2013.0062.

[7]. S. Vijayarani, and M. Vinupriya, “Performance Analysis of Canny and Sobel Edge Detection Algorithms in Image Mining”, International Journal of Innovative Research in Computer and Communication Engineering, 1(8), October 2013.

[8]. Karen Panetta, Sadaf Qazi, and Sos Again, “Techniques for Detection and Classification of Edges in Color Images”, Mobile Multimedia/Image Processing, Security and Applications, Proc. of SPIE, 6982(2), Sep 2008, doi: 10.1117/12.777703.

[9]. Theo Gevers, Arjan Gijsenij, Joost van de Weijer and Jan-Mark Geusebroek, Color in Computer Vision, Fundamentals and Applications, John Wiley & Sons, Inc., 2012.

[10]. Wang Jianwei, "An Improved Method of Color Image Edge Detection Based on One Order Gradient Operator", International Journal of Hybrid Information Technology, 6(5), pp.151-162, 2013.

[11]. Silvano Di Zenzo, "A Note on the Gradient of a Multi-Image", Computer Vision, Graphics, and Image Processing 33, pp. 116-125, 1986.

[12]. P. E. Trahanias and A. N. Venetsanopoulos, "Vector Order Statistics Operators as Color Edge Detectors", IEEE Transactions on Systems, Man, And Cyberne,Tics-Part B: Cybernetics, 26(1), pp. 135-143, February 1996. DOI: 10.1109/3477.484445

[13]. Jiangyan Xu, Linning Ye and Wang Luo, "Color Edge Detection Using Multiscale Quaternion Convolution", 2010 Wiley Periodicals, Inc., Vol. 20, pp. 354–358, 2010.

[14]. Jian Wang and Li Liu, "Specific Color-pair Edge Detection using Quaternion Convolution", 2010 3rd International Congress on Image and Signal Processing (CISP2010), Yantai, China, Oct. 2010.

[15]. Silvia Franchini, Antonio Gentile, Filippo Sorbello and Giorgio Vassallo, "Clifford Algebra based Edge Detector for Color Images", Sixth International Conference on Complex, Intelligent, and Software Intensive Systems, 2012.

[16]. Simranjit Singh Walia and Gagandeep Singh, "Color based Edge detection techniques– A review", International Journal of Engineering and Innovative Technology (IJEIT) , 3(9), March 2014.

[17]. E. Nadernejad, S. Sharifzadeh and H. Hassanpour, "Edge Detection Techniques: Evaluations and Comparisons", Applied Mathematical Sciences, 2(31), pp. 1507 – 1520, 2008.

[18]. Biswajit Mishra and Peter Wilson, "Color Edge Detection Hardware Based On Geometric Algebra", Electronic Systems Design, School of Electronics and Computer Science, University of Southampton, UK, SO17 1BJ.

Chapter Two

2. Edge Detection Techniques

2.1. Edge Detection Fundamentals

The need for edge detection algorithms as part of a vision system also has its roots in biological vision. There is compelling evidence that the very early stages of Human Visual System (HVS) contain edge sensitive cells that respond strongly when presented with edges of certain intensity and orientation. Edge detection algorithms, therefore, attempt to emulate ability present in the HVS [1, 2].

An edge in general could be a boundary between two image regions having distinct characteristics according to some feature (e.g., gray scale (luminance), color (tristimulus), and/or texture) [1]. These changes are important in the analysis of images because they often provide an indication of the physical extent of objects within the image. An operator used to detect these changes is called an edge detector. Such operators transform an image into a binary array containing ones where the magnitude of the discontinuity is significant and zeros elsewhere. The binary array obtained is usually called an edge map. This transformation is useful in image understanding systems and more other applications as illustrated in Figure 1.1 in the previous chapter. The edge map retains much of the basic structure of the image and will have less computational effort in analysis as compared to the original image [2].

There are many techniques which can be used in edge detection. These include simple differential operators, template matching, least square edge fitting, techniques based on statistical detection theory and others. A complete survey of all edge detectors is not a simple task and can even be confusing [2]. Hence, only a group of most common and useful operators will be discussed in the following sections including Roberts, Prewitt, Sobel, Compass, Log, Gradient, Canny, Vector Order Statics (VOS) and Max Gradient operators. All these have been considered to validate the performance of the proposed algorithm in this thesis using standard test images having delineated ground truths [3, 4, 5].

Linear differential operators are commonly employed in edge detection. In this method, edges are enhanced by convolving the image with a set of discrete differential operator masks. A

corresponding edge map is obtained by thresholding some function of the outputs of these masks. One of the differential operators used is the gradient; and approximately calculated by convolving the image with two masks that measure the pixels luminance change in any two orthogonal directions. The sum of the squares of the masks output is a measure of the gradient magnitude squared (shown later in Eq.2-1). Roberts has used 2x2 masks to compute the luminance difference across the diagonals while Prewitt and Sobel have used 3x3 masks to measure the difference in the horizontal and vertical directions. Another differential operator which has been used in edge enhancement is the Laplacian operator. The Laplacian method searches for the zero crossings in the second derivative of the image to find edges. In general, all of the linear differential operators have the advantage of using simple mathematical formulas which require short computation time. Their major disadvantage is their sensitivity to noise. One method to improve the performance of differential operators in the presence of noise is to increase the masks size. This can be noticed comparing the performances of the Roberts and Sobel operators [2].

Most edge detection solutions exhibit limited performance in the presence of images containing real world scenes that is, images that have not been carefully controlled in their illumination, size and position of objects and contrast between objects and background. The impacts of shadows, occlusion among objects and parts of the scene and noise to mention just a few on the results produced by an edge detection solution are often significant. Consequently, it is common to precede the edge detection stage with preprocessing operations such as noise reduction and illumination correction without losing the detail information available in the image [1].

2.1.1. Classification of Edges

Edges in images can have completely different causes due to the geometrical and photometric conditions within a scene. Different types of edges are outlined in Figure 2.1. Edges can be distinguished into the following five classes [2]:

- **Object edges or orientation edges**: Arise from a discontinuity of the vector normal to a continuous surface.
- **Reflectance edges**: Arise from a discontinuity of the reflectance of object surfaces by a change of surface material.

- **Illumination edges or shadow edges**: Arise from a discontinuity of the intensity of the incident lighting.
- **Specular edges or highlight edges**: Arise from a special orientation between the light source, the object surface, and the observer and are due to material properties.
- **Occlusion edges**: Arise from boundaries between an object and the background as seen by the observer. Occlusion edges do not represent a physical discontinuity in the scene. They exist due to a special viewing position.

In many areas of digital image processing, a classification of edges is necessary and/or advantageous. For example, only orientation edges, reflectance edges and illumination edges should be matched in stereo vision. Specular edges and occlusion edges should not be matched because their occurrence in the image depends on the viewing position of both cameras and they do not represent the identical physical locus in the scene. Illumination edges should not be matched if motion analysis is applied. The classification of edges by their physical origin is difficult or even impossible in gray scale images. Color image processing can provide a solution to this problem at least in part [2]

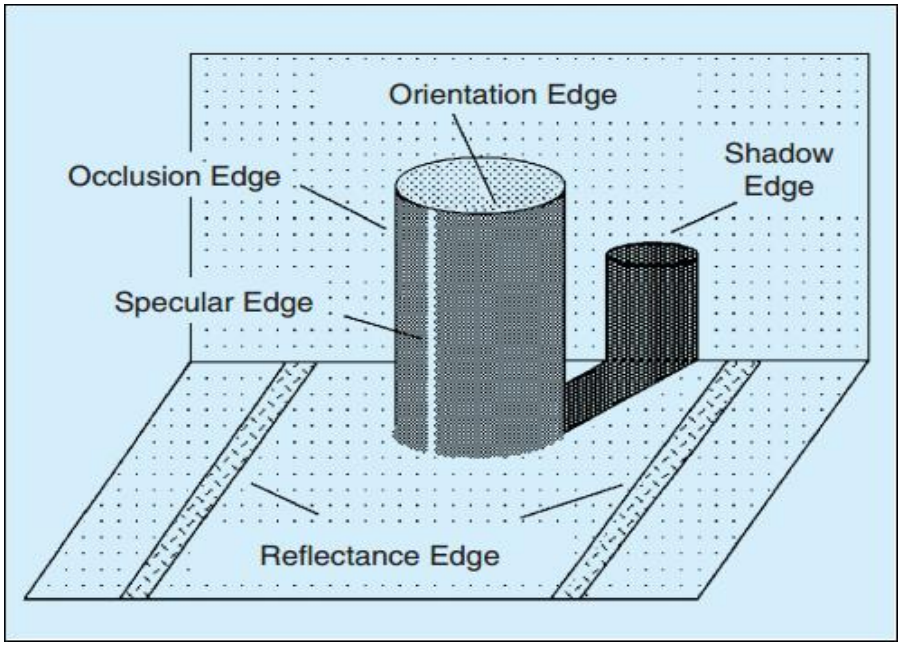


Figure 2.1 Different types of edges in a representation of a scene [6].

2.1.2. Edge Models

An edge in the given image can be either Step edge, where the image intensity abruptly changes from one value on one side of the discontinuity to a different value on the opposite side, or Line Edges, where the image intensity abruptly changes value but then returns to the starting value within some short distance [6]. However, Step and Line edges are rare in real images. Because of low frequency components or the smoothing introduced by most sensing devices, sharp discontinuities rarely exist in real signals. Illustration of these edge shapes is shown in Figure 2.2 [1, 6].

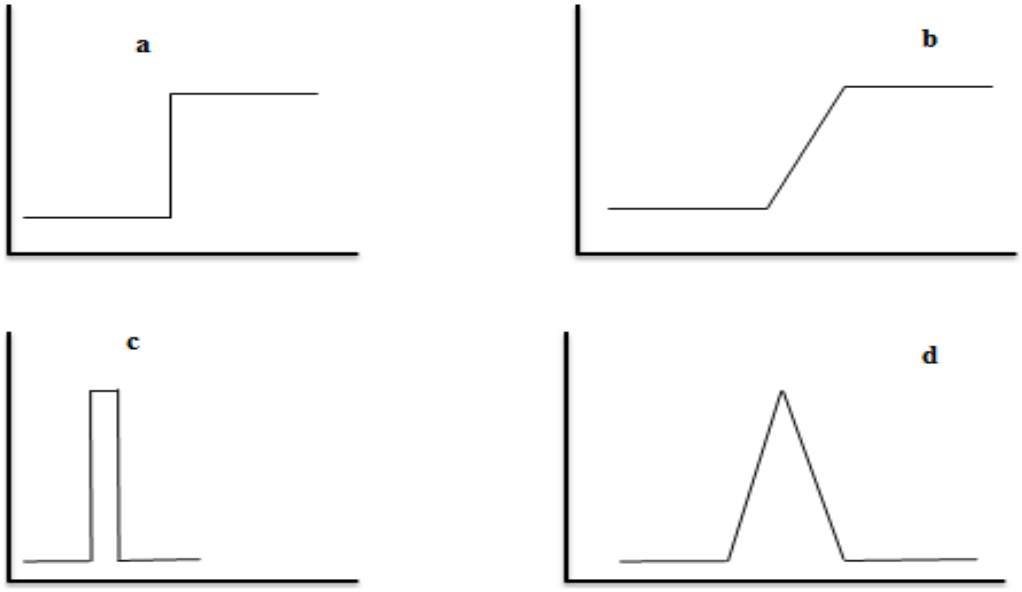


Figure 2.2 One dimensional edge profiles (a) Step Edge (b) Ramp Edge (c) Line Edge and (d) Roof Edge.

In Figure 2.3 below, an ideal edge pixel and the corresponding gradient vector are shown. In the indicated pixel the intensity changes from 0 to 255 (considering 8-bit) at the direction of the gradient. The magnitude of the gradient indicates the strength of the edge. If we calculate the gradient at uniform regions, we end up with a 0 vector which means there is no edge pixel. In natural images we usually do not have the ideal discontinuity or the uniform regions as in the Figure 2.3 and we process the magnitude of the gradient to make a decision to detect the edge pixels. The elementary processing is applied for a threshold. If the gradient magnitude is larger than the threshold, the given pixel is decided to be an edge pixel [7].

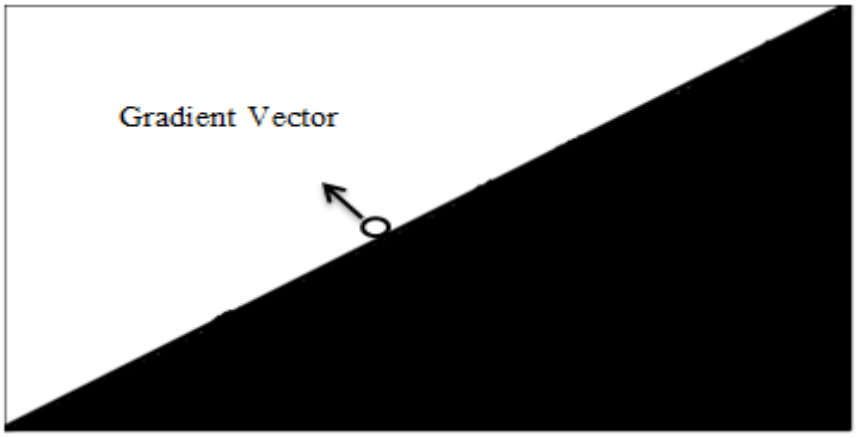


Figure 2.3. The gradient vector and an edge pixel. The circle indicates the location of the edge pixel and the arrow is the gradient vector.

In general, an edge pixel can be described by using two important features, an edge strength which is equal to the magnitude of the gradient and an edge direction which is equal to the angle of the gradient shown in Figure 2.3 [7, 8]. There are many ways to perform the edge detection operation. The next section will give us a detail review of the classical and vectorial techniques.

2.2. Edge Detectors

2.2.1. Classical Edge Detectors (Monochromatic Based)

2.2.1.1. First Order Derivative Edge Detectors (Gradient Operators)

An image gradient is a directional change in the intensity or color in an image. Image gradients may be used to extract information from images. The gradient of an image intensity function is defined at each image point as a two-dimensional (2D) vectors with the components given by the derivatives in the horizontal and vertical directions. At each image point, the gradient vector points in the direction of largest possible intensity increase and the length of the gradient vector corresponds to the rate of change in that direction as seen on Figure 2.3 above [9, 10, 11].

Edge detection methods in gray scale images usually rely on calculations of the first or second derivative along the intensity profile. The first derivative has the desirable property of being directly proportional to the difference in intensity across the edge. The magnitude of the first derivative can be used to detect the presence of an edge at a certain point in the image. The sign of the second derivative can be used to determine whether a pixel lies on the dark or on the

bright side of an edge. Moreover, the zero crossing between its positive and negative peaks can be used to locate the center of thick edges as seen in Figure 2.4 [1, 10, 11].

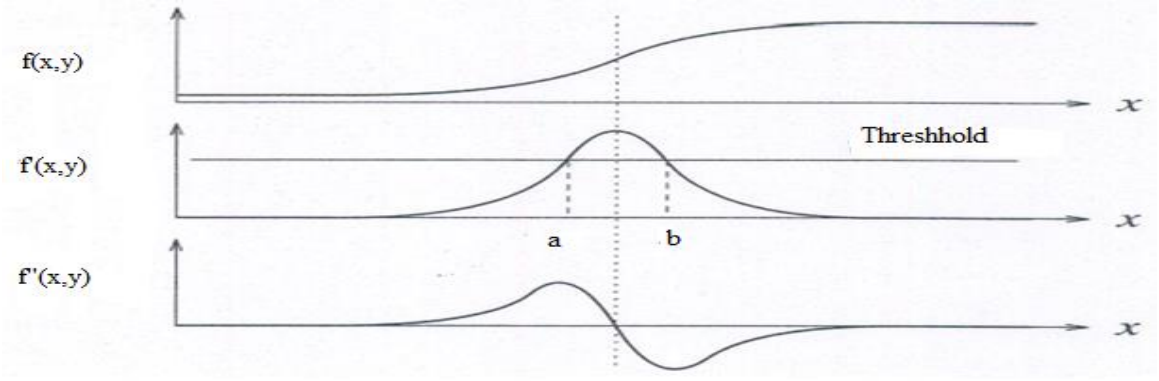


Figure 2.4. Gray scale image shown by using 1D Ramp Edge Intensity Profile (top), First Derivative Results (middle) and Second Derivative Results (bottom).

2.2.1.1.1. Roberts' Cross

Roberts Cross operator is one of the first edge detectors and was initially proposed by Lawrence Roberts in 1963 [9]. It is a differential operator [6, 9, 12]. It approximates the gradient of an image through discrete differentiation by computing the sum of the squares of the differences between diagonally adjacent pixels [9, 12, 13]. One of the most appealing aspects of this operation is its simplicity in which its kernel is small and contains only integers. However, with the speed of computers today this advantage is negligible, and the Roberts cross suffers greatly from sensitivity to noise [9]. To perform edge detection with the Roberts operator we first convolve the original image with the following two kernels [1, 9, 12].

$$\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$$

Let $\mathrm{I(x,y)}$ be a point in the original image and $G_x(\mathrm{x,y})$ be a point in an image formed by convolving with the first kernel and $G_y(\mathrm{x,y})$ be a point in an image formed by convolving with the second kernel. The gradient and direction can be defined as in Eq. 2-1and 2-2 respectively.

$$\nabla I(x,y) = G(x,y) = \sqrt{{G_x}^2 + {G_y}^2} \qquad 2\text{-}1$$

$$\theta(x,y) = arctan\,(\frac{G_y(x,y)}{G_x(x,y)} \qquad 2\text{-}2$$

2.2.1.1.2. Prewitt Filter

Prewitt edge detector is an extension of the Roberts edge detector to a 3-by-3 neighborhood. Here, the horizontal and vertical differences are computed [10, 12, 13]. The Prewitt operator is based on convolving the image with a small, separable and integer valued filter in horizontal and vertical directions. Mathematically, the operator uses two 3×3 kernels (h_x and h_y) which are convolved with the original image to calculate horizontal and vertical derivative approximations [9, 10, 12]. If we define A as the source image and G_x and G_y as horizontal and vertical derivative approximations as shown in Eq.2-3, the gradient and direction are computed as in Eq. 2-4 and 2-5 respectively [9, 14].

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad h_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\mathrm{G_x} = \mathrm{h_x} * \mathrm{A} \quad \text{and} \quad \mathrm{G_y} = \mathrm{h_y} * \mathrm{A} \qquad \text{2-3}$$

where '$*$' here denotes the 2D convolution operation.

$$G(x,y) = \sqrt{{G_x}^2 + {G_y}^2} \qquad \text{2-4}$$

$$\theta(x,y) = arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \qquad \text{2-5}$$

2.2.1.1.3. Sobel Filter

Sobel is very similar to the prewitt edge detector. It emphasizes the horizontal and vertical differences between the pixels closest to the central pixel [13]. In similar way, if we define A as the source image and G_x and G_y are horizontal and vertical derivative approximations as shown in Eq.2-6, the gradient and direction are computed as Eq. 2-7 and 2-8 respectively [9, 14].

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad h_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$G_x = h_x * A \quad and \quad G_y = h_y * A \qquad \text{2-6}$$

$$G = \sqrt{{G_x}^2 + {G_y}^2} \qquad \text{2-7}$$

$$\theta = arctan\left(\frac{G_y}{G_x}\right) \qquad \text{2-8}$$

2.2.1.1.4. Compass Mask Techniques

2.2.1.1.4.1. Kirsch Compass Masks

The idea of using horizontal and vertical masks used by the Prewitt and Sobel operators can be extended to include all eight compass directions: North (N), Northeast (NE), East (E), Southeast (SE), South (S), Southwest (SW), West (W) and Northwest (NW) [1, 2, 9]. This is done by taking a single mask and rotating it to 8 major compass orientations. The edge magnitude will be the maximum found by the convolution of each mask with the image as indicated in Eq. 2-9

$$h_{n,m} = max_{z=1,\dots,8}\left(\sum_{i=-1}^{1}\sum_{j=-1}^{1} g_{ij}^{(z)} . f_{n+i,m+j}\right) \quad \text{2-9}$$

where $g^{(1)}, g^{(2)}, g^{(3)}, g^{(4)}, g^{(5)}, g^{(6)}, g^{(7)}, g^{(8)}$ are given as E, NE, N, NW, W, SW, S, and SE respectively as shown below.

$$E = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \; NE = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \; N = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \; NW = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$W = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \; SW = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \; S = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \; SE = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

2.2.1.1.4.2. Robinson Compass Masks

Robinson compass masks can use a Sobel basic mask by rotating in 8 directions as seen below.

$$E = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \; NE = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \; N = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \; NW = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \; SW = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} \; S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \; SE = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

The only difference between Robinson and kirsch compass masks is that in Robinson, we have a standard mask but in Kirsch we can change the mask according to our own requirements.

2.2.1.2. Second Order Derivative Edge Detectors (LoG or Marr-Hildreth Edge Detector)

Some edge detection operators are instead based upon second order derivative that captures the rate of change in the intensity gradient. In the ideal continuous case, detection of zero-crossings in the second derivative captures local maxima in the gradient in order to effectively detect intensity changes (edges). The operator must be a differential operator taking second spatial derivative of the image and capable of being tuned to act at any desired scale so that large filters can be used to detect blurry shadow edges and small ones can be used to detect sharply focused fine detail in the image which is based on a scale. One such an operator is the Laplacian of Gaussian (LoG) [10]. LoG edge detector works by smoothing the image with a Gaussian Low Pass Filter (LPF) and then applying a Laplacian edge detector to the result. The LoG filter can sometimes be approximated by taking the differences of two Gaussians of different widths in a method known as difference of Gaussians (DoG) [1].

LoG is a compound operator that combines a smoothing operation using a Gaussian shaped linear phase Finite Impulse Response (FIR) filter and a differentiation operation using a discrete Laplacian. The edges are identified by the location of zero crossings which changes sign in the vicinity of maxima of the first derivative as seen earlier in Figure 2.4. Laplacian operator may detect edges as well as noise. Therefore, it may be desirable to smooth the image first by convolution with a Gaussian kernel of width α as seen in Eq. 2-10 [9].

$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left[-\frac{(x^2+y^2)}{2\sigma^2}\right]$$

$$\Delta[\mathrm{G}_\sigma(\mathrm{x},\mathrm{y}) * \mathrm{f}(\mathrm{x},\mathrm{y})] = [\Delta\mathrm{G}_\sigma(\mathrm{x},\mathrm{y})] * \mathrm{f}(\mathrm{x},\mathrm{y})] = \mathrm{LoG} * \mathrm{f}(\mathrm{x},\mathrm{y}) \qquad 2\text{-}10$$

Note that the first equality in Eq. 2-10 is because

$$\frac{d}{dt}[h(t) * f(t)] = \frac{d}{dt}\int f(\tau)h(t-\tau)d\tau = \int f(\tau)\frac{d}{dt}h(t-\tau)d\tau = f(t) * \frac{d}{dt}h(t)$$

So, we can obtain the Laplacian of Gaussian $\Delta G_\sigma(x,y)$ first and then convolve it with the input image. To do so, first consider

$$\frac{\partial G_\sigma(x,y)}{\partial x} = \frac{\partial}{\partial x} e^{-\frac{x^2+y^2}{2\sigma^2}} = -\frac{x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

and

$$\frac{\partial^2 G_\sigma(x,y)}{\partial^2 x} = \frac{x^2}{\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} - \frac{1}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = \frac{x^2-\sigma^2}{\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Note that for simplicity we omitted the normalizing coefficient $\frac{1}{\sqrt{2\pi}\sigma^2}$.

Similarly, we can get:

$$\frac{\partial^2 G_\sigma(x,y)}{\partial^2 y} = \frac{y^2-\sigma^2}{\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Now we have LoG as an operator or convolution kernel defined as

$$LoG \triangleq \Delta G_\sigma(x,y)] = \frac{\partial^2 G_\sigma(x,y)}{\partial^2 x} + \frac{\partial^2 G_\sigma(x,y)}{\partial^2 y} = \frac{x^2+y^2-2\sigma^2}{\sigma^4} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

2D LoG can be approximated by a 5x5 convolution kernel such as:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The kernel of any other sizes can be obtained by approximating the continuous expression of LoG given above. We must note that the sum or average of all elements of the kernel is zero so that the convolution result of a homogeneous region is always zero [9]. The Laplacian operator is a straightforward digital approximation of the second order derivative of the intensity.

2.2.1.3. Canny Edge Detector (Gradient Gaussian Edge Detector)

The Canny edge detector has been widely considered to be the standard edge detection algorithm from classical edge detectors. It was first introduced by John Canny for his master's thesis at Massachusetts Institute of Technology (MIT) in 1983 and still outperforms many of the newer algorithms that have been developed. Canny saw the edge detection problem as a signal processing optimization problem. So, he developed an objective function to be optimized. The steps in the canny edge detector are as follows [5]:

i. Smooth the image with a two-dimensional Gaussian.
ii. Take the gradient of the image.

iii. Perform non-maximal suppression. Non maximum suppression works by finding the pixel with the maximum value in an edge. In order to do this, the magnitude and direction of the gradient are computed at each pixel. Then for each pixel, we will check if the magnitude of the gradient is greater than one pixel away in either the positive or the negative direction of the edge. If the pixel is not greater than both, suppress it (see Figure 2.5) based on the following function indicated in Eq. 2-11[5].

$$M(x,y) = \begin{cases} |\nabla S|(x,y) & \text{if } |\nabla S|(x,y) > |\nabla S|(x',y') \text{ and } |\nabla S|(x,y) > |\nabla S|(x'',y'') \\ 0 & \text{Otherwise} \end{cases} \quad \text{2-11}$$

where $S(x,y)$ is the pixel value at position (x,y).

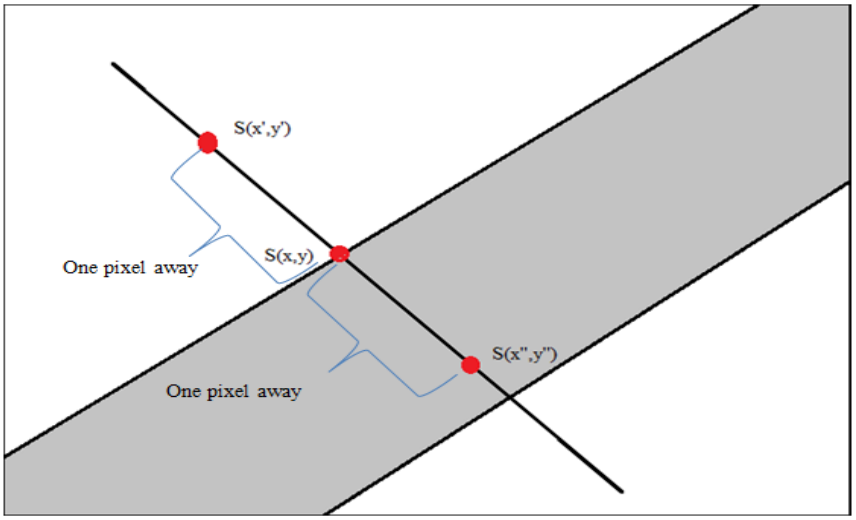


Figure 2.5. Non maximal suppression.

iv. Perform Edge thresholding. It uses two thresholds T_{low} and T_{high}. Edge pixels with values greater than T_{high} are considered as strong edge pixels whereas edge pixels with values between T_{low} and T_{high} are said to be weak pixels. Here if a pixel is connected to any pixel above T_{high}, it is considered as a strong pixel and otherwise not. Pixels completely below T_{low} are not considered as an edge pixel [5] as seen in Figure 2.6 below.

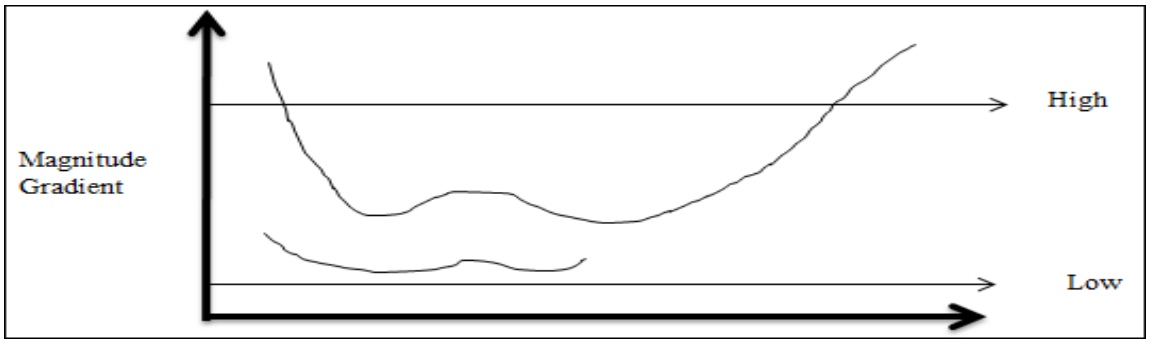


Figure 2.6. Hysteresis thresholding.

2.2.2. Vectorial Approaches

2.2.2.1. Review of Vector Valued Techniques

In this sub-section, we will present a review of techniques for the detection and classification of edges in color images. Figure 2.7 demonstrates a typical difference between edge detection in gray scale and color spaces. It takes Lena test image and applies edge detection: first considering Lena as a gray scale intensity image and then the full Lena color image. It is clear from the results that the method that considered the whole color information (in this case trinion based edge detection which will be explained in detail later) outperformed the classical approach. The accuracy in detecting these edges and the efficiency in implementing these operations are important criteria for using an algorithm in the area of computer vision and medical image analysis. Inaccuracies in edge detection directly influence the results of subsequent image analysis procedures which benefit from accurate edge detection tools [2].

Figure 2.7. Results of edge detection applied to color test image of Lena. Original color image (left), gray scale-based edge output (middle) and result using vectorial (trinion based) approach (right).

Gray scale edge detection has been given more attention compared to color image edge detection. Most of the color edge detection methods proposed in the literature are monochromatic based techniques which can be applied either to a selected color component or after transforming the colors into gray scale images using some technique. Holistic approaches that consider the inter-correlation information embedded between the color channels/bands are considered more accurate and robust [2]. Below we will review some of the vectorial edge detection schemes which exist in the literature while recently proposed approaches that showed great promises in the area are revisited in the upcoming chapters [2].

2.2.2.1.1. Operators Based on Vector Order Statistics

The general principle to these algorithms is to break an image into windows and determine if an edge exists inside each window based on statistics of the pixels inside the window and a user set threshold. There are four algorithms using this general strategy: Vector Range (VR) edge detection, Vector Dispersion (VD) edge detection, Mean Vector Range (MVR) edge detection, and Mean Vector Dispersion (MVD) [15]. Each algorithm has been explained thoroughly.

2.2.2.1.1.1. Vector Range (VR) Edge Detection

The detailed implementation of the VR edge detection algorithm is as follows:

1. A 3 by 3 pixel window is taken from the image.

2. For each pixel in the window a vector of size 3 is used to describe the color as $pixel_{m,n}RGB$.

3. A new set of scalars,$[d_0 d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8]$, is calculated for each pixel by determining the Euclidean distance between a given $pixel_{m,n}RGB$ and all other $pixel_{m,n}RGB$ in the window.

   a. The pixels in a window are indexed such that the top left is $(1,1)$ and bottom right is (3,3)

   $(1,1)$ $(2,1)$ $(3,1)$
   $(1,2)$ $(2,2)$ $(3,2)$
   $(1,3)$ $(2,3)$ $(3,3)$

   The Euclidean distance between a pair of vectors is given using below Eq. 2-12

$$\left|pixel_{1,1}RGB - pixel_{1,2}RGB\right| = \sqrt{(pixel_{1,1}R - pixel_{1,2}R)^2 + (pixel_{1,1}G - pixel_{1,2}G)^2 + (pixel_{1,1}B - pixel_{1,2}B)^2} \quad 2\text{-}12$$

b. This results in 9 distances: $[d_0 d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8]$, from 9 pixel distance calculations $|pixel_{1,1}RGB - pixel_{1,1}RGB|, |pixel_{1,1}RGB - pixel_{1,2}RGB|, |pixel_{1,1}RGB - pixel_{1,3}RGB|$, ... and so on in which one of the differences is always 0. i.e. $|pixel_{1,1}RGB - pixel_{1,1}RGB|$

c. The 9 distance scalars are summed to produce a single scalar for the given pixel. i.e. $d_{1,1} = d_0 + d_1 + d_2 + d_3 + d_4 + d_5 + d_6 + d_7 + d_8$

4. This process is repeated for each pixel in the window to get a single scalar for each pixel. The initial 3x3x3 matrix has been transformed into a 3x3x1 matrix.

$$\begin{bmatrix} pixel_{1,1}RGB & pixel_{1,2}RGB & pixel_{1,3}RGB \\ pixel_{2,1}RGB & pixel_{2,2}RGB & pixel_{2,3}RGB \\ pixel_{3,1}RGB & pixel_{3,2}RGB & pixel_{3,3}RGB \end{bmatrix} \rightarrow \begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,1} & d_{3,2} & d_{3,3} \end{bmatrix}$$

5. The new 3x3x1 matrix is reshaped into a 9x1 array; the index corresponds to the pixel number.

$$\begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,1} & d_{3,2} & d_{3,3} \end{bmatrix} \rightarrow \begin{bmatrix} d_{1,1} \\ d_{2,1} \\ d_{3,1} \\ d_{1,2} \\ d_{2,2} \\ d_{3,2} \\ d_{1,3} \\ d_{2,3} \\ d_{3,3} \end{bmatrix}$$

6. The original 3x3x3 window is reshaped in a manner related to the reshaping of the 3x3x1 matrix to result in a 9x3 matrix.

$$\begin{bmatrix} pixel_{1,1}RGB & pixel_{1,2}RGB & pixel_{1,3}RGB \\ pixel_{2,1}RGB & pixel_{2,2}RGB & pixel_{2,3}RGB \\ pixel_{3,1}RGB & pixel_{3,2}RGB & pixel_{3,3}RGB \end{bmatrix} \rightarrow \begin{bmatrix} pixel_{1,1}RGB \\ pixel_{2,1}RGB \\ pixel_{3,1}RGB \\ pixel_{1,2}RGB \\ pixel_{2,2}RGB \\ pixel_{3,2}RGB \\ pixel_{1,3}RGB \\ pixel_{2,3}RGB \\ pixel_{3,3}RGB \end{bmatrix}$$

7. The 9x1 array is sorted in ascending order and the same rearrangement of indices is applied to the first dimension of the 9x3 matrix i.e.

$$d_{1,1} \leq \ d_{2,2} \leq \ d_{3,3} \leq d_{1,2} \leq d_{2,1} \leq \ d_{3,1} \leq \ d_{1,3} \leq \ d_{2,3} \leq \ d_{3,2}$$

$$\begin{bmatrix} d_{1,1} \\ d_{2,1} \\ d_{3,1} \\ d_{1,2} \\ d_{2,2} \\ d_{3,2} \\ d_{1,3} \\ d_{2,3} \\ d_{3,3} \end{bmatrix} \rightarrow \begin{bmatrix} d_{1,1} \\ d_{2,2} \\ d_{3,3} \\ d_{1,2} \\ d_{2,1} \\ d_{3,1} \\ d_{1,3} \\ d_{2,3} \\ d_{3,2} \end{bmatrix} = \begin{bmatrix} pixel_{1,1}RGB \\ pixel_{2,1}RGB \\ pixel_{3,1}RGB \\ pixel_{1,2}RGB \\ pixel_{2,2}RGB \\ pixel_{3,2}RGB \\ pixel_{1,3}RGB \\ pixel_{2,3}RGB \\ pixel_{3,3}RGB \end{bmatrix} \rightarrow \begin{bmatrix} pixel_{1,1}RGB \\ pixel_{2,2}RGB \\ pixel_{3,3}RGB \\ pixel_{1,2}RGB \\ pixel_{2,1}RGB \\ pixel_{3,1}RGB \\ pixel_{1,3}RGB \\ pixel_{2,3}RGB \\ pixel_{3,2}RGB \end{bmatrix}$$

8. VR is calculated by finding the Euclidian distance between the first pixel and the last pixel resulting from the ordering of the sort. i.e.

$$VR = |pixel_{3,2}RGB - pixel_{1,1}RGB|$$

9. If the VR is above a user set threshold then the window contains an edge and the center pixel of the window is set to 1 to represent an edge at that location.

10. The process is then repeated on a new 3 by 3-pixel window of the image shifted 1 row or 1 column of pixels from the previous 3 by 3 window until all 3 by 3 windows have been evaluated resulting in an edge map of the image.

2.2.2.1.1.2. Minimum Vector Range (MVR) Edge Detection

MVR edge detection is similar with VR edge detection. It considers the difference between the last k vectors and the first vector rather than the difference between the first and last vectors from the sort described above for the VR implementation. MVR edge detection tends to be better than VR edge detection because it alleviates the sensitivity of VR to long tailed noise.

$$VR1 = \left|pixel_{3,2}RGB - pixel_{1,1}RGB\right|$$

$$VR2 = |pixel_{2,3}RGB - pixel_{1,1}RGB|$$

$$\vdots$$

$$VR_k = |pixel_{r,s}RGB - pixel_{1,1}RGB|$$

$$MVR = \min\{VR_1, VR_2, \dots, VR_k\}$$

2.2.2.1.1.3. Vector Dispersion (VD) Edge Detection

VD edge detection is based on some linear combination of linear operations, averaging is done on the first 3 vectors to generate a new first pixel i.e.

$$pixel_{one}RGB = \frac{pixel_{1,1}RGB + pixel_{2,2}RGB + pixel_{3,3}RGB}{3}$$

The same processes are then carried out as in VR edge detection except rather than using the first vector from the sort one uses the newly calculated $pixel_{one}RGB$ i.e.

$$VR = |pixel_{3,2}RGB - pixel_{one}RGB|$$

This adjustment also tends to be an improvement over VR edge detection in that it is less susceptible to short tailed noise.

2.2.2.1.1.4. Minimum Vector Dispersion (MVD) Edge Detection

MVD is the combination of MVR and VD edge detection. The MVR edge detection process is carried out with the change that rather than find the minimum distance between the last k pixels and the first pixel, the minimum difference between the last k pixels and the average of the first 3 pixels is calculated. i.e.

$$VR1 = \left|pixel_{3,2}RGB - pixel_{one}RGB\right|$$

$$VR2 = |pixel_{2,3}RGB - pixel_{one}RGB|$$

$$\vdots$$

$$VR_k = |pixel_{r,s}RGB - pixel_{one}RGB|$$

$$MVR = \min\{VR_1, VR_2, \dots, VR_k\}$$

References

[1]. Oge Marques, Practical Image and Video Processing Using Matlab, A John Wiley and Sons, Inc., Publication, ISBN 978-0-470-04815-3

[2]. A. Koschan and M. Abidi, "Detection and Classification of Edges in Color Images," Signal Processing Magazine, Special Issue on Color Image Processing, 22(1), pp. 64-73, 2005.

[3]. Karen Panetta, Sadaf Qazi, and Sos Again, "Techniques for Detection and Classification of Edges in Color Images," Mobile Multimedia/Image Processing, Security, and Applications 2008, Proc. of SPIE Vol. 698269820W, 2008, doi: 10.1117/12.777703.

[4]. Dawit Assefa Haile, The 1D and 2D Localized Hartley Transforms their Parallel Implementation and applications; Color Image Analysis Using Quaternions and Trinions, PhD Dissertation, London, Ontario, Canada, 2007.

[5]. E. Nadernejad, S. Sharifzadeh and H. Hassanpour, "Edge Detection Techniques: Evaluations and Comparisons," Applied Mathematical Sciences, 2(31), pp. 1507 – 1520, 2008.

[6]. N. Senthilkumaran1 and R. Rajesh, "Edge Detection Techniques for Image Segmentation – A Survey of Soft Computing Approaches," International Journal of Recent Trends in Engineering, 1(2), May 2009.

[7]. G. T. Shrivakshan, C. Chandrasekar, "A Comparison of various Edge Detection Techniques used in Image Processing," IJCSI International Journal of Computer Science Issues, 9(5), September 2012.

[8]. Ikram Escandar Abdou, "Quantitative Methods of Edge Detection, Image Processing Institute, University of Southern California," Los Angeles, California 90007, US, July 1978.

[9]. Indra Kanta Maitra, Sanjay Nag and Samir K. Bandyopadhyay, "A Novel Edge Detection Algorithm for Digital Mammogram," International Journal of Information and Communication Technology Research, 2(2), February 2012.

[10]. Li Bin, Mehdi Samiei yeganeh, "Comparison for Image Edge Detection Algorithms," IOSR Journal of Computer Engineering (IOSRJCE), 2(6), July-Aug. 2012.

[11]. Sukhjinder Kaur and Archna Mahajan, “Edge Improvement of Clustered Soybean Seeds by using Sobel Filter,” International Journal of Scientific & Engineering Research, 5(5), May-2014.

[12]. Poonam Dhankhar and Neha Sahu, “A Review and Research of Edge Detection Techniques for Image Segmentation,” International Journal of Computer Science and Mobile Computing, IJCSMC, 2(7), pp.86 – 92, July 2013.

[13]. Slawomir Bogumil Wesolkowski, Color Image Edge Detection and Segmentation: A Comparison of the Vector Angle and the Euclidean Distance Color Similarity Measures, MSc Dissertation, Waterloo, Ontario, Canada, 1999.

[14]. Linhua Zhang, Xiuli Mao, Chuanbin Zhou and Ping Yu, “Improved HIS Model with Application to Edge Detection for Color Image,” Journal of Computers, 7(6), June 2012.

[15]. P. E. Trahanias and A. N. Venetsanopoulos, “Vector Order Statistics Operators as Color Edge Detectors, IEEE Transactions on Systems,” Man, And Cyberne,Tics-Part B: Cybernetics, 26(1), February 1996.

Chapter Three

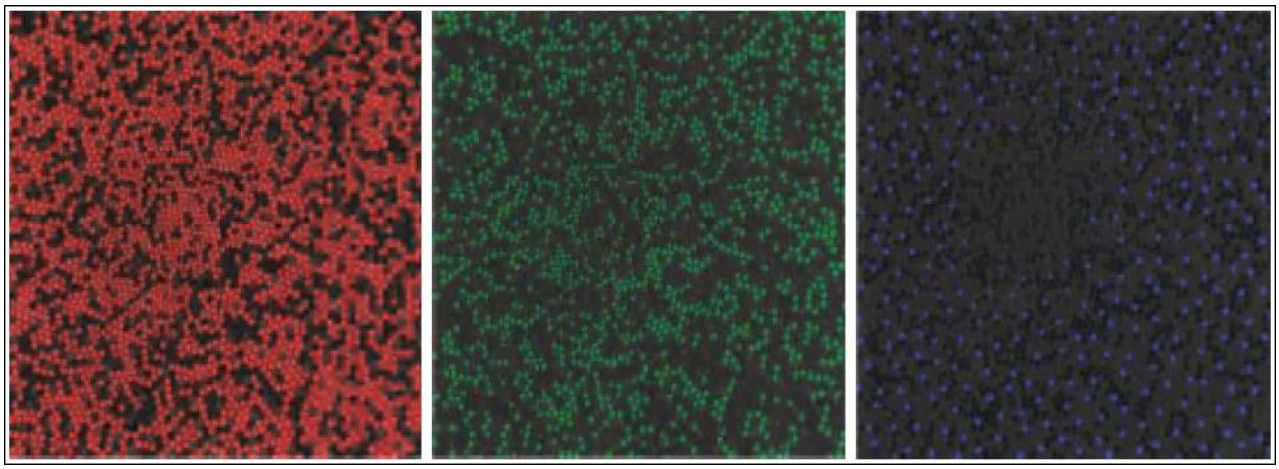
3. Medical Image Processing and Color Fundamentals

3.1. Colors

3.1.1. Introduction

Visual information is our most natural source of information and communication. With increasing production, use and exploitation of digital visual information, a visual overflow will occur, and hence demands are urgent for the automatic understanding of visual information [1].

Humans perceive color as a result of light having wavelengths in the region of 400 nm to 700 nm being projected upon the retina. Color is the brain's reaction to a specific visual stimulus of light of a certain wavelength. The human retina has three color photoreceptor cone cells, each of which responds to incident radiation with somewhat different spectral response curve as seen in Figure 3.1 and Figure 3.2 below. These three broad spectral bands roughly correspond to what humans perceive as red, green and blue light. The rod is the fourth type of photoreceptor cell present in the retina. The signals from these color sensitive cells (cones) together with rods (sensitive to intensity only) combine in the brain to give sensation of different colors [1, 2].



a b c

Figure 3.1 Cone mosaic at the central fovea showing (a) L-cones (b) M-cones and (c) S-cones. The area shown is approximately 0.3×0.3 mm and is rod-free. The labeling in red, green, and blue refers to the spectral region where the cones have their maximum sensitivity [1].

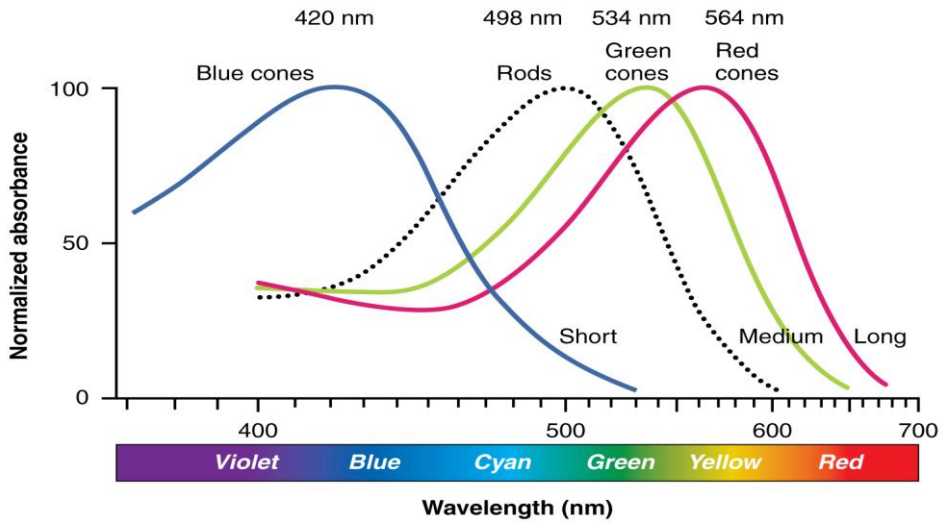


Figure 3.2 Normalized human photoreceptor absorbances for different wavelengths of light [1].

3.1.1.1. Color Space (Models)

A color space is a method by which we can specify, create and visualize color. Color may be defined by its attributes of brightness, hue and purity for humans. A computer may describe a color using the amounts of R, G and B phosphor emission required to match a color. A printing press may use the reflectance and absorbance of cyan (C) magenta (M) and yellow (Y) inks on the printing paper to generate a specific color (black ink is also used for gray tones). Thus, a color is usually specified using three co-ordinates or parameters (except for printing although technically speaking black is not considered a color but the absence of a color). These parameters describe the position of the color within the color space being used. The following section describes the color models or spaces which are used in this thesis. Each color space (RGB and opponent color models) are briefly discussed together with its properties and computational complexity [1, 2].

Color vectors in each of these spaces differ from one another. This difference means that two colors in one space being separated by one distance value would be separated by a different distance value in another space. A color represented in one space can be changed to another spatial representation by performing some linear or non-linear transformation. RGB space is the physical sensor-based color space and all other color spaces are derived from it. It is still used more than any other space today in image processing applications due to its direct relationship with the physical world. In this thesis, RGB and HSI will be discussed in a bit detail. There are

many other spaces including CMY, YIQ, YUV, HSI or its generalized form HSV, CIELAB, CIELUV, rgb (known as the normalized RGB space), c1c2c3, l1l2l3, YCbCr, and the Angle Space. In this thesis, RGB and HSI/HSV will be discussed in a bit detail [1, 2, 3].

3.1.1.1.1. RGB Color Model

The RGB space is used most frequently in computer graphics and image processing applications. A color in this space is represented by a triplet of values typically between zero and one and is usually scaled by 255 for 8 bit representation. Each color can be broken down into its relative intensity in the three primaries corresponding to the spectral response of one of the three types of cones present in the human eye. The space is easily represented as a three dimensional cube where each axis represents the strength of the color in one of the three primaries as seen in Figure 3.3 [1, 2].

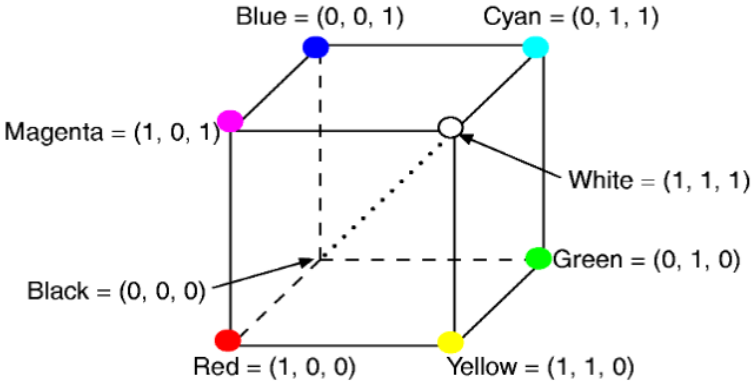


Figure 3.3 The RGB color space.

3.1.1.1.2. Opponent Color Spaces

One of the properties of RGB is that the values of the three channels are highly correlated. Decorrelating the RGB color space leads to an opponent color space. The opponent color theory started at about the year 1500 when Leonardo da Vinci concluded that colors are produced by the mixture of yellow and blue, green and red, and white and black. This opponent color theory has been completed by Edwald Hering concluding that the working of the eye is based on the three kinds of opposite colors.

Color vision starts with the absorption of light in three cone types. Responses arising from these cones are combined in retinal ganglion cells to form three opponent channels; one achromatic

(black - white) and two chromatic channels (red - green and yellow - blue). Retinal ganglion cells sendoff pulse like signals through the optic nerve to the visual cortex, where the perception of color eventually takes place [1, 4, 5]. In this regard, one the color models is denoted as opponent color space which is used in this thesis and can be computed by simply rotating the RGB color system as seen in Eq.3-1, 3-2 and 3-3 and in Figure 3.4

$$O_1 = \frac{R - G}{\sqrt{2}} \tag{3-1}$$

$$O_2 = \frac{R + G - 2B}{\sqrt{6}} \tag{3-2}$$

$$O_3 = \frac{R + G + B}{\sqrt{3}} \tag{3-3}$$

Note that O_1 roughly corresponds to the red-green channel, O_2 corresponds to the yellow-blue channel and O_3 corresponds to the intensity channel.

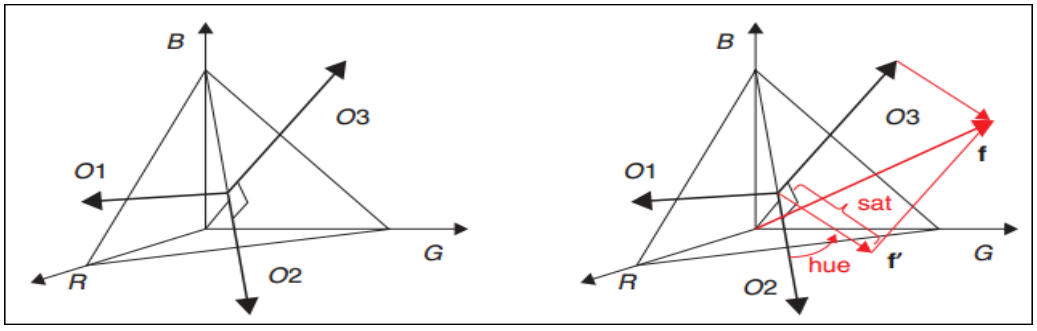


Figure 3.4. The opponent axes (left) and the relation between the opponent and hue-saturation-intensity coordinate system (right) [1].

Projection of f on O_3 is the intensity. Let f' be the projection of f on the $O_1 - O_2$ plane; then its length is the saturation and the angle in the plane is the hue. Within the context of coordinate transformations, the hue-saturation-intensity transformation is computed by performing a polar coordinate transformation on the opponent color plane formed by $O_1$and O_2, according to the transformation depicted in Figure 3.4. Note that it is more correct to talk about saturation strength in this transformation because the saturation has not been normalized by the intensity as in Eq. 3-4 [1, 4, 5]

$$\begin{pmatrix} H \\ S \\ I \end{pmatrix} = \begin{pmatrix} arctan\dfrac{O_1}{O_2} \\ \sqrt{O_1^2 + O_2^2} \\ O_3 \end{pmatrix} = \begin{pmatrix} arctan\left(\dfrac{\sqrt{3}(R-G)}{R+G-2B}\right) \\ \sqrt{\dfrac{4}{6}(R^2+G^2+B^2-RG-RB-GB)} \\ \dfrac{R+G+B}{\sqrt{3}} \end{pmatrix} \quad 3\text{-}4$$

3.2. Image Processing

3.2.1. Fundamentals

Digital image processing means processing of images which are digital in nature by a digital computer. Digital image processing is motivated by three major applications. The first application is improvement of pictorial information for human perception, this means enhancing the quality of the image so that the image will have a better look. The second important application is for autonomous machine applications, particularly for medical image analysis or quality control in assembly automation and many such applications in industries. A third application is for efficient storage and transmission. If we want to store an image on a computer, then this image will need certain amount of disk space and band width during transmission.

In the area of medicine, digital image processing techniques are used in many different applications. One is to determine the location of a tumor say based on MRI scans or detect the contour of the brain tumor, as demonstrated in Figure 3.5, for example. Cancerous tissues will be contoured in mammogram images which show the presence of affected tissues. Digital image processing will be helpful to detect the formation of cancers. In Ultrasonography also, we can get ultrasonic images which are used to study the growth of a baby while the baby is in the mother's womb and this helps doctors to monitor the health of the fetus. There are many more applications of digital image processing in the medical world which have been reported in different literatures.

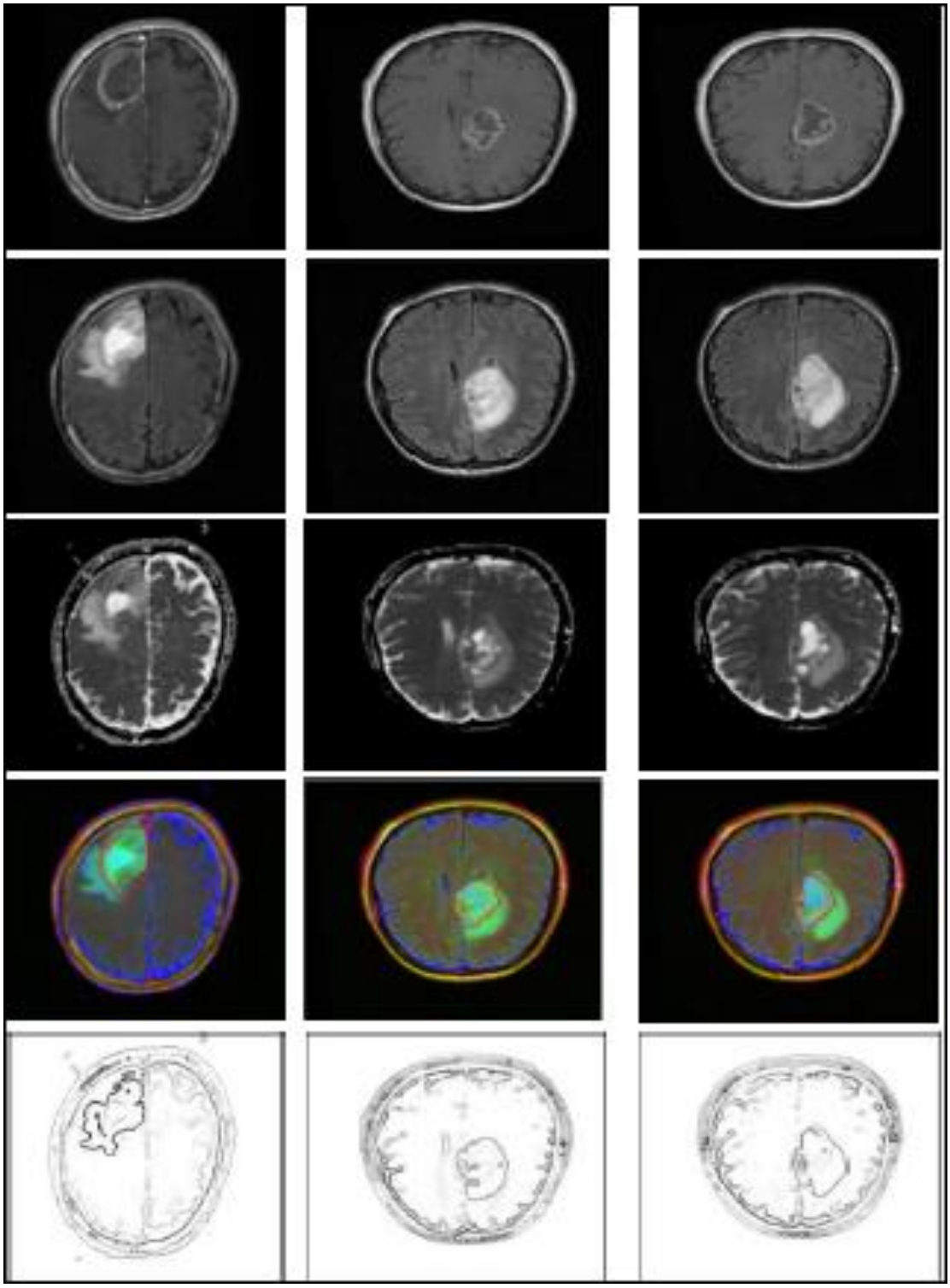


Figure 3.5 Row 1 to 3 are original T1, T2-FLAIR and ADC images respectively of different MRI slices of two patients containing contrast enhanced glioma tumors; row 4 contains the respective combined color images (red contour shows tumor delineations by a radiation oncologist) and row 5 contains the corresponding edge showing tumor [6].

3.2.1.1. *Digital Images*

Digital images are comprised of a set of points or picture elements usually referred to as pixels stored as an array of numbers. They are spatial data indexed by two spatial coordinates mostly with variables x and y referring to the horizontal and vertical axes of an image. Pixel value represents the color or intensity of each pixel and the placement of the pixels within the matrix correspond to their placement within the image. If more than one value is required to encode pixel information, the image is often represented by a multidimensional matrix. For example, an RGB encoding of an image would contain three (3) matrices, one each for R, G and B intensities. In other terms each pixel represented in the matrix has a value that is encoded as either a scalar (in the case of gray-scale) or a vector (in the case of color) as shown Figure 3.6.

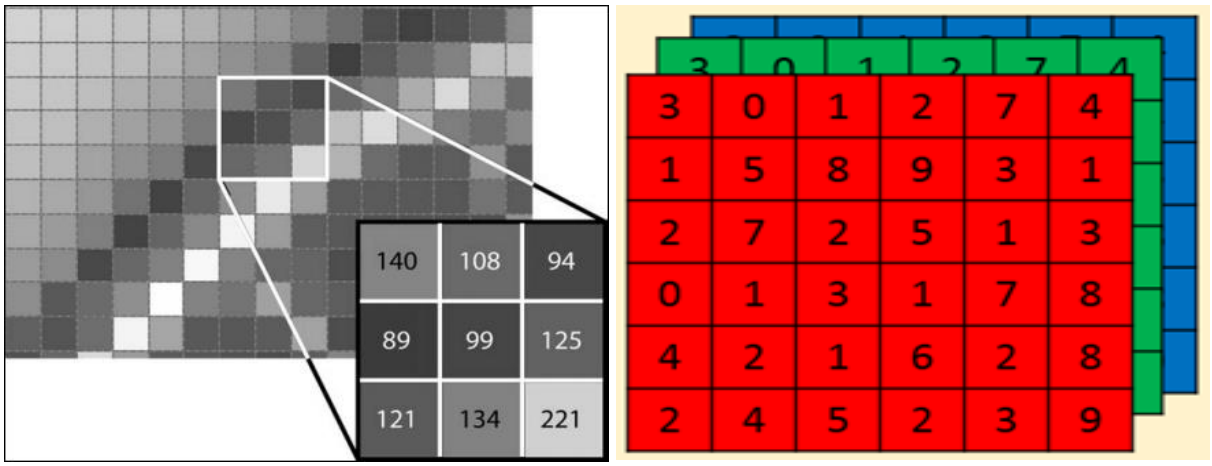


Figure 3.6 Gray scale image representation (left) and Color image representation (right).

3.2.1.2. Color Image Processing

The subject of color image processing has gained attention because color images convey more information about objects in a scene than gray scale images and this information can be used to further refine the performance of an imaging system. The multichannel nature of a color image, however, adds considerable complexity to the processing system [5].

3.3. Method of Retinal Image Acquisition

Fundus (Latin for "bottom") is an anatomical term referring to the portion of an organ opposite from its opening. The fundus of the eye is the interior surface of the eye, opposite the lens which includes the retina, optic disc (OD), macula and fovea, and posterior pole (i.e. the fundus) [7]. A

fundus camera or retinal camera is a specialized low power microscope with an attached camera designed to photograph the interior surface of the eye, including the retina, OD, macula, and posterior pole. It is used by optometrists, ophthalmologists, and trained medical professionals for monitoring progression of a disease, diagnosis of a disease (combined with retinal angiography), or in screening programs, where the photos can be analyzed later [7, 8].

The retina can be photographed directly as the pupil is used as both an entrance and exit for the fundus camera's illuminating and imaging light rays. The patient sits at the fundus camera with their chin in a chin rest and their forehead against the bar. An ophthalmic photographer focuses and aligns the fundus camera. A flash fires as the photographer presses the shutter release, creating a color fundus photograph. Ophthalmologists use these retinal photographs for diagnosis, treatment, and subsequent follow ups of eye diseases.

Compared to other retinal imaging techniques, for instance Optical Coherence Tomography (OCT) and hyper spectral imaging techniques, fundus photography generally needs a considerably cheaper instrument. Figure 3.7 shows a typical fundus camera where patient is sitting to have fundus photography by the operator. It has the advantage of availing the image to be examined by a specialist at another location and/or time as well as providing color photo documentation for future reference [8, 9].

Figure 3.7 Fundus camera [9].

3.3.1. Optical Principles of Fundus Camera

The optical principles of the fundus camera are similar with principles of monocular indirect ophthalmoscopy. Light generated from either viewing lamp or the electronic flash is projected through a set of filters and onto a round mirror. This mirror reflects the light up into a series of lenses to focus the light. A mask on the uppermost lens shapes the light into a doughnut shape. The doughnut shaped light is reflected onto a round mirror with a central aperture, exits the camera through the objective lens, and proceeds into the eye through the cornea. Figure 3.8 shows a general schematic diagram of the light rays' path in a fundus camera [9, 10].

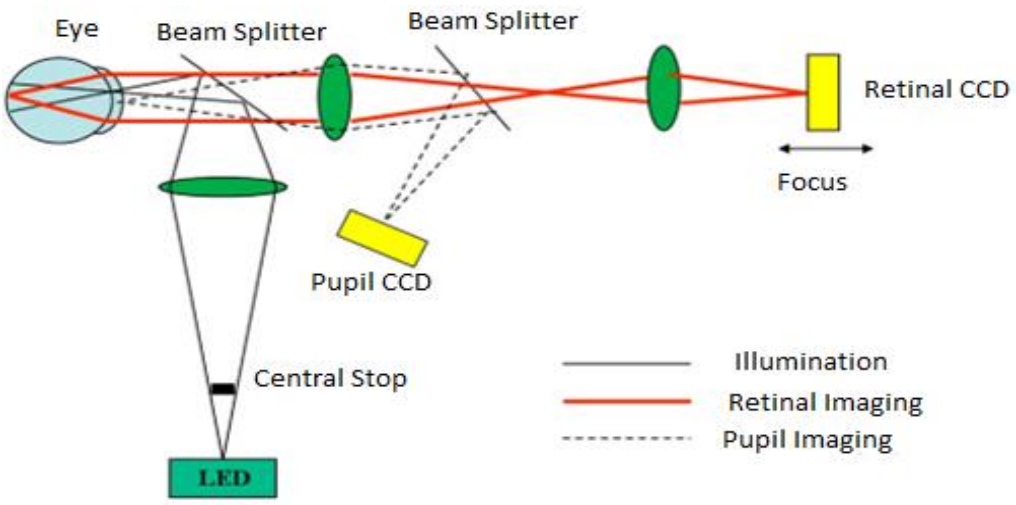


Figure 3.8 Schematic diagram of a fundus camera optical system [8].

3.4. Diabetic retinopathy

Diabetes has become one of the rapidly increasing health threats worldwide. The diabetes may cause abnormalities in the retina (diabetic retinopathy), kidneys (diabetic nephropathy), and nervous system (diabetic neuropathy) [11]. Fundus imaging has an important role in diabetes monitoring and a candidate for non-invasive screening of diabetes since occurrences of retinal abnormalities are common with serious consequences.

Diabetic retinopathy is a complication of diabetes, causing abnormalities in the retina, and in the worst case, blindness. Typically, there are no salient symptoms in the early stages of diabetes, but the number and severity predominantly increase with time. The diabetic retinopathy typically begins as small changes in the retinal capillary. The first detectable abnormalities are

microaneurysms which are local distensions of the retinal capillary and which cause intra-retinal hemorrhage (Figure 3.9(c)) when ruptured. In time, the retinal edema and hard exudates (Figure 3.9 (b)) are followed by the increased permeability of the capillary walls. Hard exudates are lipid formations leaking from these weakened blood vessels. This state of the retinopathy is called non-proliferative diabetic retinopathy. However, if the abovementioned abnormalities appear in the central vision area (macula), it is called diabetic maculopathy. As the retinopathy advances, the blood vessels become obstructed which causes micro-infarcts in the retina. These micro-infarcts are called soft exudates (Figure 3.9 (a)). When significant number of soft exudates (> 6) or intra-retinal micro-vascular abnormalities are encountered, the state of the retinopathy is defined as pre-proliferative diabetic retinopathy [11].

The pre-proliferative diabetic retinopathy can quickly turn into proliferative diabetic retinopathy when extensive lack of oxygen causes the development of new fragile vessels. This is called neovascularization which is a serious state threatening eyesight. The field of eyesight can be obstructed by a hemorrhage to the vitreous body which is a common cause of blindness. The neovascularization can tear retina and when it is located near the center of macula, it can cause the loss of sight. Due to the progressive nature of the retinopathy, regular monitoring is needed after diagnosis. This will need medical experts to identify the different stages and/or an automatic image processing method must be developed. For this purpose, an effective edge detection which will contour the different features in the given fundus image plays a key role.

Figure 3.9 Abnormal findings in the eye fundus caused by DR: (a) Soft exudates (b) Hard exudates and (c) hemorrhages [courtesy of St. Paul's Hospital Millinium Medical College, ophthalmology department].[8].

References

[1]. Theo Gevers, Arjan Gijsenij, Joost van de Weijer and Jan-Mark Geusebroek, “Color in Computer Vision, Fundamentals and Applications,” A John Wiley & Sons, Inc., Publication, 2012.

[2]. Slawomir Bogumil Wesolkowski, Color Image Edge Detection and Segmentation: A Comparison of the Vector Angle and the Euclidean Distance Color Similarity Measures, MSc Dissertation University of Waterloo, Waterloo, Ontario, Canada, 1999.

[3]. Kiran Jot Singh and Monika Aggarwa, “Edge Detection Algorithm for Color Images using Logical Operations,” International Journal of Computer Science & Communication Networks, 2(1), 83-86.

[4]. Rao Muhammad Anwer, David V´azquez, and Antonio M. L´opez, “Opponent Colors for Human Detection,” Springer-Verlag Berlin Heidelberg 2011, LNCS 6669, pp. 363–370, 2011.

[5]. T. Marko, T. Jurij, “Colour spaces, Project for the Digital signal processing course,” Faculty of electrical engineering University of Ljubljana, Slovenia, 2012.

[6]. Dawit Assefa, Harald Keller, and David A. Jaffray, “Signal Analysis Of Multi-Parametric Mr Images In Higher Order Fourier Spaces,” International Journal of Computational Bioscience, 4(1), 2013.

[7]. Kade Mahesh K and Kashid Nilesh S, “Review on Fundus Image Acquisition Techniques with Data base Reference to Retinal Abnormalities in Diabetic Retinopathy,” International Journal of Computer Applications (0975 – 8887), 68(8), April 2013.

[8]. P. J. Saine, Fundus Photography: “Fundus Camera Optics,” Ophthalmic Photographers Society. Available online: https://www.opsweb.org/page/fundusphotography, Website Referenced on Jan. 10, 2014.

[9]. R., Besenczi, et al, "A review on automatic analysis techniques for color fundus photographs", omputational and Structural Biotechnology Journal, 14(2), pp. 371-384, 2016.

[10]. M. Kade, N. Kashid, “A review on automatic analysis techniques for color fundus photographs,” Computational and Structural Biotechnology Journal, 68(8), April 2013.

[11]. T. Kauppi, V. Kalesnykiene, J. Kamarainen, L. Lensu, I. Sorri, A. Raninen, R. Voutilainen, H. Uusitalo, H. Kalviainen and J. Pieti, “DIARETDB1 diabetic retinopathy database and evaluation protocol,” pp. 1-18, 2010.

Chapter Four

4. Trinion and Quaternion Based Color Edge Detection

4.1. Quaternions

4.1.1. Introduction to Quaternion

In this section, we will see about signals and images that have vector valued samples (that is samples with three or more components) and their processing techniques using Fourier transforms based on four dimensional hyper complex numbers or quaternions (Q). Quaternion Fourier Transforms (QFT) have applications in processing of complex signals exploiting their symmetry properties [11, 14, 15].

4.1.1.1. Definitions

Quaternions are one of the four existing normed division algebras over the real numbers. Classically denoted by H in honor of Sir W. R. Hamilton who discovered them in 1843, they form a non-commutative algebra. A quaternion Q is a four dimensional (4D) hyper complex number and has a Cartesian form given by Eq. 4-1 below [11, 12, 14].

$$Q = a + ib + jc + kd \tag{4-1}$$

where a, b, c and d are four real numbers and the elements $\{i, j, k\}$ are square roots of -1 and obey the following rules as in Eq. 4-2 [5, 7, 13].

$$\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{ijk} = -1,\ \mathrm{ij} = \mathrm{k},\ \mathrm{jk} = \mathrm{i},\ \mathrm{ki} = \mathrm{j} \text{ and } \mathrm{ji} = -\mathrm{k},\ \mathrm{kj} = -\mathrm{i},\quad \mathrm{ik} = -\mathrm{j} \tag{4-2}$$

Generally, a quaternion Q has one real part and three orthogonal imaginary parts [1, 11, 14]. It is usually represented in the form shown in Eq. 4-3 below.

$$Q = S(Q) + V(Q) \tag{4-3}$$

where $S(Q) = a$ is called the scalar part of Q and $V(Q) = ib + jc + kd$ is called the vector (or imaginary) part.

4.1.1.2. Properties

Quaternion algebra is distinct from both real and complex algebra. For example, multiplication is not commutative in the quaternion space as shown in Eq.4-2. Here it is interesting to note the following exponential property in the quaternion space [11, 12, 13].

$e^{ib+jc} \neq e^{ib}e^{jc}$, this shows that operations which are trivial in both real and Fourier spaces are not so obvious in the case of quaternions [10, 11, 12].

The addition, subtraction and multiplication of two quaternions P and Q are defined as in Eq. 4-4 and 4-5 [1, 13, 17].

$$P \pm Q = (a_1 + ib_1 + jc_1 + kd_1) \pm (a_2 + ib_2 + jc_2 + kd_2) \quad 4\text{-}4$$
$$= (a_1 \pm a_2) + i(b_1 \pm b_2) + j(c_1 \pm c_2) + k(d_1 \pm d_2)$$

$$PQ = -P.Q + PxQ \quad 4\text{-}5$$
$$= (a_1a_2 + b_1b_2 + c_1c_2 + d_1d_2) + (a_1b_2 + a_2b_1 + c_1d_2 - d_1c_2)i$$
$$+ (a_1c_2 + c_1a_2 - b_1d_2 - d_1b_2)j + (a_1d_2 + d_1a_2 + b_1c_2 + c_1b_2)k$$

Any can be written as $Q = \| Q \| e^{\mu\emptyset}$

$\| Q \| = \sqrt{QQ_c} = \sqrt{a^2 + b^2 + c^2 + d^2}$ is called the amplitude or modules and $\mu = \frac{V(Q)}{|V(Q)|}$ is called Eigen axis and $\emptyset = \arctan\left(\frac{|V(Q)|}{S(Q)}\right)$, $0 \leq \emptyset < \pi$ is called Eigen angle (phase), where $\mu \epsilon Q$, $S(\mu) = 0$, $|\mu| = 1$ and $\mu^2 = -1$.

The conjugate of a quaternion $Q = a + ib + jc + kd$ is $Q_c = a - ib - jc - kd$ and if $\left|\left|Q\right|\right| = 1$, then Q is called unit quaternion. If the real part of Q, that is, a is equal to 0, then Q is called pure quaternion. If $\left|\left|Q\right|\right| = 1$ and $a = 0$, then Q is called unit pure quaternion [1, 6, 8, 13].

4.1.2. Representation of Images in the Quaternion Space

Images can be represented in different color spaces. The previous chapter discussed about the different color models and their computational advantages. There are different ways to map a color image in the quaternion space. One way which has been adopted in the current study is to map the three component color pixels to a unit pure quaternion [5, 11, 12, 13]. For an RGB color image $h(x, y)$, it can be mapped as $h(x, y) = R(x, y)i + G(x, y)j + B(x, y)k$.

4.1.3. The Quaternion Fourier Transform

QFT is defined similar with standard Fourier transform of 2D functions. The non-commutative nature of quaternion multiplication allows for the definition of different types of QFT. The dominant ones in literature are listed below [1, 11]:

The QFT of Type I is given by:

$$Q(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-j2\pi ux} h(x,y)\, e^{-k2\pi vy}\; dxdy$$

where x and y represent spatial variables while u and v are the wavenumbers (inverse of wavelengths) in the respective directions and $h(x,y)$ is a quaternion. The inverse type I QFT differs from the forward only at the sign of the exponents in the two exponential terms.

The QFT of Type II is given by:

$$\mathrm{Q(u,v)} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-2\pi\mu(ux+vy)} h(x,y)\, dxdy$$

where $h(x,y)$ is generally a quaternion valued image function and μ is a unit pure quaternion, such that the product μ^2= -1. The formula for the inverse type II QFT differs from the forward one only at the sign of the exponent in the exponential term. The FT is a special case in which $\mu = i$. Generally, the parameter μ is chosen arbitrarily [11, 12, 13].

The discrete variants of type I and II QFTs have been suggested in different literatures [11] as an auxiliary transform to reduce the computational complexity of the complex DFT [2, 3, 4].

Quaternion uses holistic approach to extract important features in color images. Most color images, however, have three components and the extra fourth dimension in quaternions creates redundancy in representing such images, so the proposed technique uses trinion for representation in three spaces and corresponding integral transforms to efficiently analyze three component color images.

4.2. Trinions

4.2.1. Introduction to Trinions

Most color images have three components and the three component numbers (trinions) are best suited to analyze those kinds of images [14, 15]. In the case of quaternions, we often have better flexibility because of the fourth dimension. However, as most color images have three components such an extra dimension does not help much and creates redundancy. The three-dimensional trinions can help alleviate this redundancy [11, 15, 16].

4.2.1.1. Definition

A trinion number is defined with one real and two imaginary components. A trinion number can be written as:

$t = a + ib + jc$

where a, b, c are real numbers and i, j are operators satisfying the following multiplication rules as seen in Eq. 4-6 below.

$$ii = j, \qquad jj = -i \;\; and \; ij = ji = -1 \tag{4-6}$$

The three base elements $\{1, i, j\}$ of trinions form an abelian group where 1 is the multiplicative identity element.

4.2.1.2. Properties

Distinct from quaternions, trinions with the above structure form a commutative ring. Trinions are commutative and associative both under addition and multiplication.

Any trinion number t can be expressed as the sum of a real part and a vector part as [16]:

$t = S(t) + V(t)$

where $S(t) = a$ is the real part and $V(t) = ib + jc$ is the vector part.

It can also be written in the following form:

$t = |t|(cos\,\varphi + \mu sin\,\varphi)$

where $|t| = \sqrt{a^2 + b^2 + c^2}$, $\mu = \frac{V(t)}{|V(t)|}$ and $\varphi = tan^{-1}\left(\frac{|V(t)|}{S(t)}\right)$, $0 < \emptyset < \pi$ are the amplitude (modulus), the eigen axis and eigen angle (phase) respectively. When $|t| = 1$, t is said to be a unit trinion and when $a = 0$, it is called a pure trinion.

The addition or subtraction and multiplication of two trinions t_1 and t_2 are defined as in Eq. 4-7 and 4-8 respectively [1, 13, 16].

$$t_1 \pm t_2 = (a_1 + ib_1 + jc_1) \pm (a_2 + ib_2 + jc_2) = (a_1 \pm a_2) + i(b_1 \pm b_2) + j(c_1 \pm c_2) \quad 4\text{-}7$$

$$t_1 t_2 = (a_1 a_2 - b_1 c_2 - c_1 b_2) + i(a_1 b_2 + b_1 a_2 - c_1 c_2) + j(a_1 c_2 + b_1 b_2 + c_1 a_2) \quad 4\text{-}8$$

Trinion exponentiation is rather complicated. It has been shown that trinion exponentiation can be written in terms of hyper geometric functions as follows [13].

Let $t = a + ib + jc$. Then the following is obviously true, $e^t = e^a e^{ib} e^{jc}$.

Analogous to the quaternion, trinions are identified as two-dimensional complex space C_i^2 . For example if $t = a + ib + jc$, then $t = S_1 + S_2 j$, where $S_1 = ib \in C_i^1$ and $S_2 = c - ai \in C_i^1$. Similarly, t can be identified as two-dimensional complex space C_j^2 [10, 11, 12].

4.2.2. Trinion Based Edge Detection

4.2.2.1. Introduction

Most approaches that are suggested in the literature for holistic processing of color images depend on the color intensity information to extract features for different applications as illustrated in Figure 4.1 below (top part). However, other than the intensity, color images have at least two other attributes (the hue and the saturation). For applications such as edge detection, texture analysis and pattern recognition all three information are vital. The proposed scheme in this thesis tries to address the above issues by adopting recently introduced holistic ways of color representation in opponent color space to map in the trinion or pure quaternion space. The method keeps the inter correlation information that is embedded across the different channels/bands of color images. The proposed method uses the three component trinions to map the three color components of a color image to a single number (trinion) and applies higher order FT in the new space. A trinion is defined with one real and two imaginary components as discussed in the above section. The advantage of the use of trinions is that it allows a holistic

representation of color images and the respective FT allow their analysis as one entity without the need to separate the different bands [6] as shown in Figure 4.1 below (bottom part).

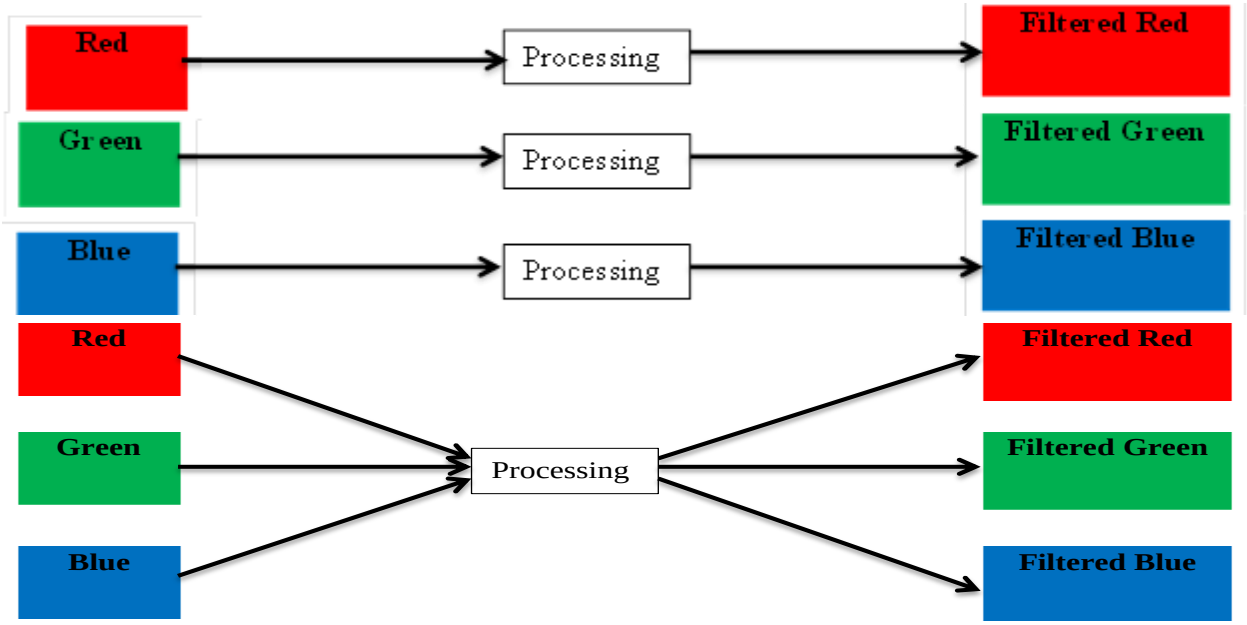


Figure 4.1 Serial (Marginal) approaches (top) and Vectorial methods (bottom).

4.2.3. Color Space Selection and Transformation

The color perceived by a human eye can be defined as a linear combination of R, G and B. However, there are other color spaces which would be more useful in applications than just RGB color space. In this regard, transforming images from RGB space to a different color space can be very beneficial. One commonly used such space is the opponent color space which uses H, S and I where H refers to hue, S refers to saturation and I refers to intensity/luminance/lightness. This color space is believed to be most likely the kind of color perceived by our eyes. In our case, each component of the RGB color space is normalized to its maximum and mapped to the HSL utilizing a color space transformation. The RGB to HSL conversion was needed as the later color space is believed to be closer to the human visual perception and hence is potentially helpful for our edge detection scheme as stated in opponent color space [11, 12]. The original RGB image was converted to HSL color using the formula shown previously in Eq. 3-4 given above in section 3.1.1.1.2

4.2.3.1. Representation of Color Image in the Trinion Space

The converted RGB color image is mapped to a trinion as: $h(x, y) = H + iS + jL$ in the case of HSL color space. Previous studies have shown that the order of the mapping to a trinion doesn't affect the image analysis [16].

4.2.3.2. Color Gradient

Image derivatives are essential to describe the local structure in images. Being essential operations, image derivatives have been used in vast majority of computer vision applications and medical image analysis including basic operations including edge detection, feature extraction, optical flow and more complex applications such as shape from shading, image segmentation, and object detection [18].

Image derivatives for gray value images are obtained by smoothing and differentiation in both the x- and y- directions. In our notation, for an image E, the intensity gradient is denoted by $\nabla E = (E_x, E_y)$ with its magnitude $E_w = |\nabla E| = \sqrt{E_x^2 + E_y^2}$. Each of the gradient components is obtained by convolving each channels of the image E with the Gaussian derivative functions vertically and horizontally. Here, the spectral parameters of the Gaussian functions are fixed, the Gaussian is centered at a fixed wavelength and has a fixed standard deviation (spectral bandwidth) and uses a first-order derivative comparing the yellow to the blue part of the spectrum. The total edge strength due to the color information in an image will be obtained by taking the magnitude [18].

4.2.3.3. The Trinion Fourier Transform

Defining a Fourier type of transform in the trinion space is somehow difficult. One of the reasons is the non-invertibility of trinions. Particularly, we need those properties satisfied when we want to define the inverse transform. Since trinion exponentiation is not simple, we cannot try defining our transform using an exponential kernel like the standard Fourier transform. For that reason, a different approach has been used in a previous study [12, 16].

Two working definitions for the TFT have been suggested. The TFT of type I and its Inverse Trinion Fourier Transform (ITFT) are given by:

$$\mathrm{T(u,v)} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x,y)\left(\cos\left(2\pi(ux+vy)\right) - \mu_1 \sin\left(2\pi(ux+vy)\right)\right) dxdy$$

$$\mathrm{h(x,y)} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} T(u,v)\left(\cos(2\pi(ux+vy)) + \mu_2 sin\left(2\pi(ux+vy)\right)\right) dudv$$

where $h(x,y)$ is generally a trinion valued image function, μ_1 is a unit pure trinion and μ_2 is a trinion such that the product $\mu_1\mu_2 = -1$. The choice of $\mu_1$and μ_2 is arbitrary. For instance if the choice of $\mu_1 = \frac{(i-j)}{\sqrt{2}}$, then $\mu_2 = \frac{(-1-i+j)}{\sqrt{2}}$ to keep the relationship $\mu_1\mu_2 = -1$ satisfied.

There exists also type II TFT in the literature [12, 16]. However, many operations of interest including convolutions and correlations are shown to be easier using the type I TFT than type II TFT. Hence the former has been adopted in the current study. So the discrete type I TFT and its inverse were computed and adopted in this study as shown in Eq. 4-9 and 4-10.

$$T(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} h(x,y)\left(cos\ (2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)) - \mu_1\, sin(\ 2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right))\right) \quad 4\text{-}9$$

$$\mathrm{h(x,y)} = \sum_{\mathrm{u=0}}^{\mathrm{M-1}}\sum_{\mathrm{v=0}}^{\mathrm{N-1}} \mathrm{T(u,v)}\left(\cos\ (2\pi\left(\frac{\mathrm{ux}}{\mathrm{M}}+\frac{\mathrm{vy}}{\mathrm{N}}\right)) + \mu_2 \sin(\ 2\pi\left(\frac{\mathrm{ux}}{\mathrm{M}}+\frac{\mathrm{vy}}{\mathrm{N}}\right))\right) \quad 4\text{-}10$$

where $M \times N$ is the total number of voxels (vectors) present in the selected region of interest (window) of the original image, $u = 0, 1, \ldots, N-1$ and $v = 0, 1, \ldots, M-1$ are the discrete frequencies along the horizontal and vertical directions respectively [7, 15].

4.2.4. Edge Detection in the Trinion Space

Once each color pixel represented in the HSV or HSL space is mapped to a trinion, a voxel by voxel analysis is carried out by computing the TFT based on a translating localizing window of size 3 by 3. On the resulting 3 by 3 trinion valued matrix a second order feature was computed and assigned to the central voxel value. For the purpose of edge detection, the feature F was computed using Eq. 4-11 [9, 15].

$$F = Metric\left(\frac{\gamma}{2*\beta*gamma\left(\frac{1}{\gamma}\right)} exp\ \left(-\left|\frac{p(x,y)^{\gamma}}{\beta}\right|\right)\right) \quad 4\text{-}11$$

where $0 < \gamma \leq 2$*, is the shape parameter,* $\beta > 0$ *is the scale parameter and the variance of this distribution is* $\frac{\beta^2 * gamma\left(\frac{3}{\gamma}\right)}{gamma\left(\frac{1}{\gamma}\right)}$

where $p(x, y)$ is the square amplitude or power of the TFT, and $Metric$ is a second order statistical feature (variance, standard deviation, etc.) and $gamma$ is an interpolating factorial function. The above step is repeated across all pixels that are included in the color image under consideration and the resulting matrix is plotted as an edge map. Note that all discrete TFT calculations were performed using a repeated use of the standard 2D Fast Fourier Transform (FFT). The results

4.2.4.1. Algorithm - Summary

The color components in RGB color space are highly correlated; so the first step in our edge detection scheme utilizes a color space transformation. Each component of a RGB image is first normalized to its maximum and then the normalized RGB image is transformed to the opponent color space. The reason for RGB to HSI conversion was based on explanations in chapter three suggesting the superiority of this color space in many useful image processing applications. The conversion formula has been presented in section 3.1.1.1.2. The color components are highly decorrelated at this stage and color gradient technique has been used to give emphasis to describe the local structures in the image.

Following this step, the resulting color image in the HSL space is mapped to a trinion space. It has already been shown previously that the order should not affect many image processing applications including texture analysis, pattern recognition and other spectral analysis applications [12, 16]. Hence the conversion doesn't alter the color wavenumber spectra. The proposed edge detection scheme in the current study relies on robust features (texture, pattern) generated through holistic analysis of color pixels, and hence the issue of order of the mapping is not relevant, i.e., the order of the mapping from the HSI to the trinion space shouldn't affect our color edge detection. Then a spatially localized analysis was carried out on a pixel-by-pixel basis by computing the TFTs over a translating localizing window of size 3×3. On the resulting 3×3 trinion valued matrix, a second order feature was computed and assigned to the central pixel value. For the purpose of edge detection, the feature *F* was computed using the formula given in Eq. 4-11. Figure 4.2 shows the general flow chart of the proposed edge detection algorithm.

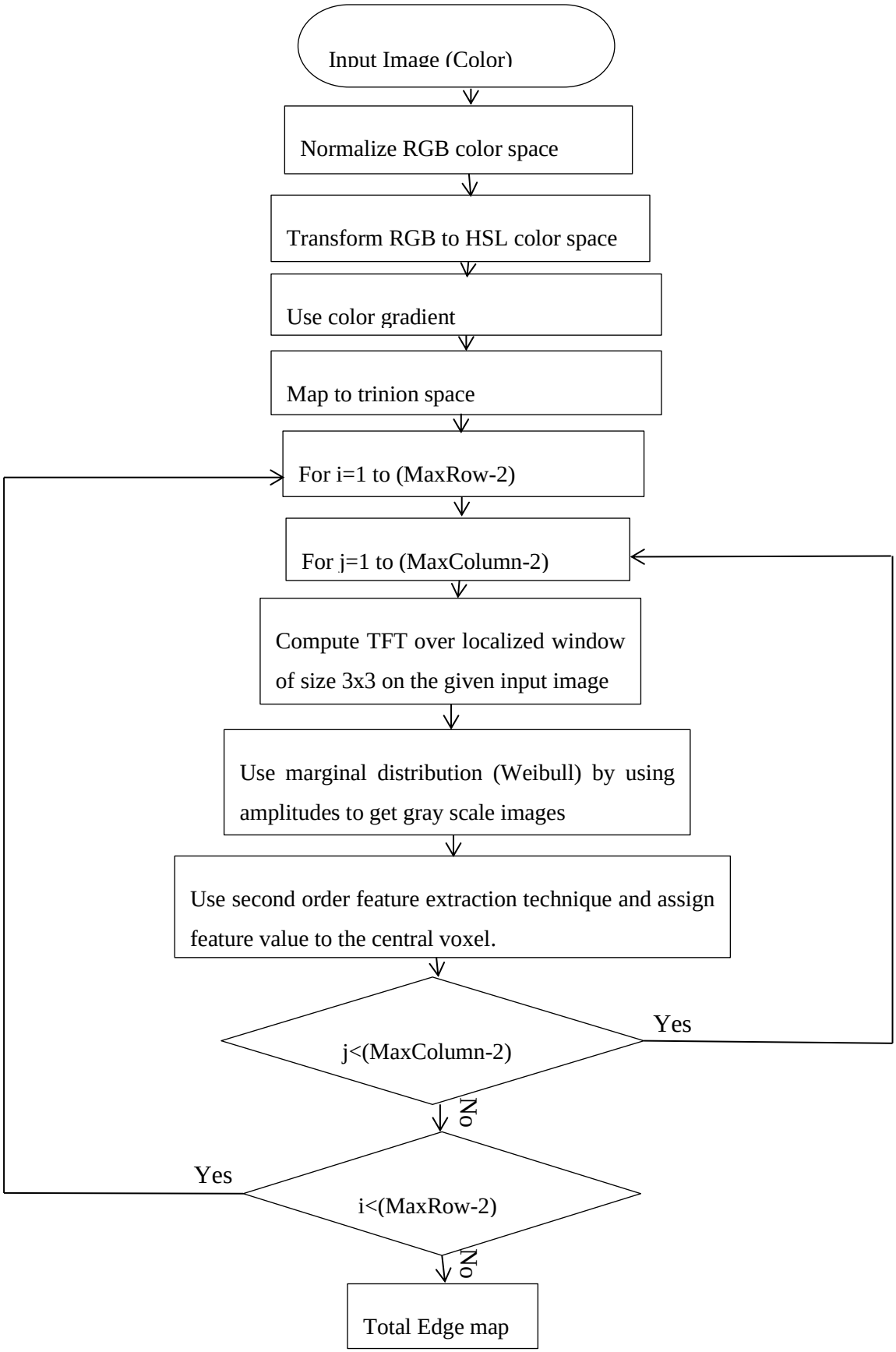


Figure 4.2 Flow chart of the proposed edge detection Algorithm.

References

[1]. Vikas R. Dubey, “Quaternion Fourier Transform for Colour Images,” International Journal of Computer Science and Information Technologies, 5(3), pp. 4411-4416, 2014.

[2]. M.I. Khalil, “Applying Quaternion Fourier Transforms for Enhancing Color Images,” I.J. Image, Graphics and Signal Processing, 2012, 2, pp. 9-15, DOI: 10.5815/ijigsp.2012.02.02.

[3]. Mawardi Bahri and Surahman, “Discrete Quaternion Fourier Transform and Properties,” Int. Journal of Math. Analysis, 7(25), 2013, pp. 1207 – 1215.

[4]. Özlem N, Subakan and Baba C. Vemuri, “A Quaternion Framework for Color Image Smoothing and Segmentation,” Int J Comput Vis (2011) 91: 233–250 DOI 10.1007/s11263-010-0388-9.

[5]. B. D.Venkatramana Reddy and T. Jayachandra Prasad, “Colour -Texture Image Segmentation using Hypercomplex Gabor Analysis,” Signal & Image Processing: An International Journal(SIPIJ) 1(2), December 2010.

[6]. P. Denis, P. Carre and C. Fernandez-Maloigne, “Spatial and spectral Quaternionic approaches for Colour Images,” Computer Vision and Image Understanding, 107(1), pp. 74-87, Jul 2007.

[7]. E. Jebamalar Leavline and D. Asir Antony Gnana Singh, “On Teaching Digital Image Processing with MATLAB,” American Journal of Signal Processing 2014, 4(1): 7-15, DOI: 10.5923/j.ajsp.20140401.02.

[8]. Nicolas Le Bihan and Stephen J. “Sangwine, Quaternion Principal Component Analysis of Color Images,” International Conference on Image Processing, October 2003 DOI: 10.1109/ICIP.2003.1247085.

[9]. Dawit Assefa, Harald Keller, and David A. Jaffray, “Signal Analysis of Multi-Parametric MR Images in Higher Order Fourier Spaces,” International Journal of Computational Bioscience, 4(1), 2013.

[10]. Todd A. Ell and Stephen J. “Sangwine, Hypercomplex Fourier Transforms of Color Images,” IEEE Transactions on Image Processing, 16(1), January 2007.

[11]. Lianghai Jin, Enmin Song, Lei Li and Xiang Li, “A Quaternion Gradient Operator for Color Image Edge Detection,” 2013 IEEE, pp. 3040-3044.

[12]. Dawit Assefa Haile, The 1D and 2D Localized Hartley Transforms their Parallel Implementation and applications; Color Image Analysis Using Quaternions and Trinions, PhD Dissertation, The University of Western Ontario, London, Ontario, Canada, 2007.

[13]. Soo-Chang Pei, Jian-Jiun Ding and Ja-Han Chang, “Efficient Implementation of Quaternion Fourier Transform, Convolution and Correlation by 2-D Complex FFT,” IEEE Transactions on Signal Processing, 49(11), November 2001.

[14]. Todd A. Ell, Nicolas Le Bihan and Stephen J. Sangwine, “Quaternion Fourier Transforms for Signal and Image Processing,” Digital Signal and Image Processing Series, John Wiley & Sons, Inc., 2014.

[15]. Dawit Assefa et al., “Local quaternion Fourier transform and color image texture analysis,” Signal Processing 90 (2010), pp. 1825–1835 doi:10.1016/j.sigpro.2009.11.031.

[16]. Dawit Assefa et al., “The trinion Fourier transform of color images,” Signal Processing 91 (2011), pp. 1887–1900 doi:10.1016/j.sigpro.2011.02.011.

[17] J. Wang and L. Liu, "Specific color-pair edge detection using quaternion convolution," 2010 3rd International Congress on Image and Signal Processing, pp. 1138-1141, Ocy 2010.

[18]. Theo Gevers, Arjan Gijsenij, Joost van de Weijer and Jan-Mark Geusebroek, Color in Computer Vision, Fundamentals and Applications, A John Wiley & Sons, Inc., Publication, 2012.

Chapter Five

5. Results and Discussion

5.1. Data Sets

The following publicly available online and local data sets have been used in the current work in order to assess the efficacy of the proposed color edge detection scheme:

- DIARETDB0: 130 color fundus images of which 20 are normal and 110 contain signs of the diabetic retinopathy (hard exudates, soft exudates, microaneurysms, hemorrhages and neovascularization) [1, 2].
- DIARETDB1: 89 color fundus images of which 84 contain at least mild non-proliferative signs (Microaneurysms) of the diabetic retinopathy, and 5 are considered as normal which do not contain any signs of the diabetic retinopathy according to all experts who participated in the evaluation. [3, 4].
- Digital Retinography System (DRS) fundus images taken from St. Paul's Hospital Millennium Medical College in Ophthalmology Department: 11 fundus images have been chosen from hundreds of images, where others are not considered for testing purpose due to eye and camera alignment problem, reflectance and patient eye blinking effects during camera flashing.

5.2. Experimental Results

5.2.1. Classical edge detection techniques (First and Second order) vs proposed method

In order to check the efficacy of the proposed edge detection scheme and compare it against classical techniques, testing has been done on synthetic, pseudo random color pattern and real color scenes. Figure 5.1 contains three test images presented here for demonstration: a synthetic test pattern, a natural scene as well as another pseudo random color pattern. Visually it is very clear that the proposed edge detection technique that makes use of weibull distribution function computed in the HSL color space based on TFT performed much more superior to the classical techniques. This is quite expected as most of the classical techniques make use of just intensity information to compute the edges while the proposed technique is a holistic approach that incorporates the entire color information.

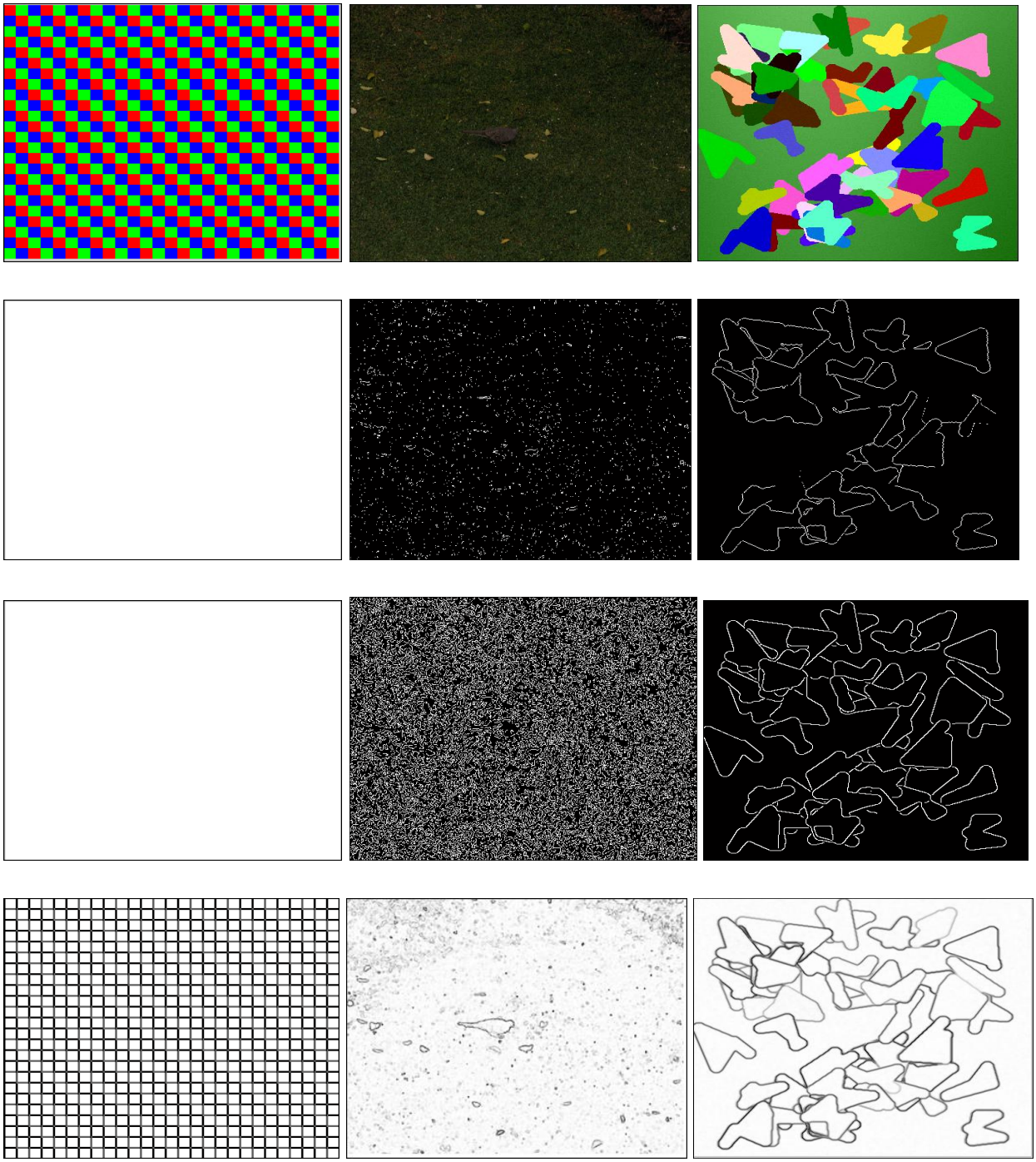
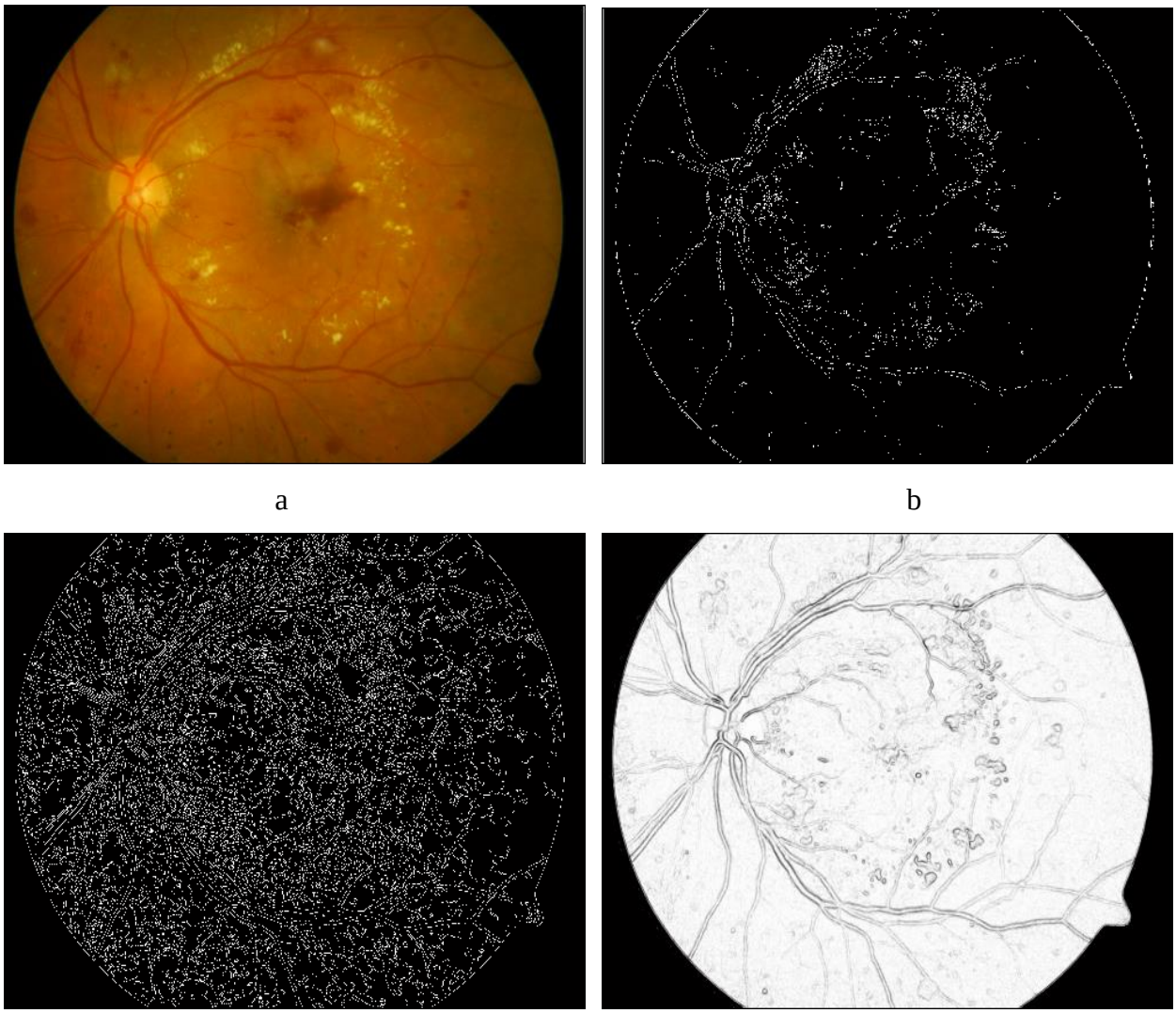


Figure 5.1 Original test images (row 1), Sobel edge map (row 2), Canny edge map (row 3) and proposed technique (row 4)

The test image presented in Figure 5.2 contains a color image of the human retina captured by a fundus camera. The optical disc (OD), blood vessel, exudates (hard and soft) and other structures are of interest while detecting edges on retinal images. It is clear from the figure that the proposed edge detection technique effectively segmented out those most dominant edges in the retinal images while the classical techniques were far much inferior and almost completely failed in detecting any of the edges.



a b

c d

Figure 5.2 Original DR image (a), Sobel edge map (c), Canny edge map (b) and proposed technique (d)

5.2.2. Vectorial approaches (the max gradient, VOS) vs the proposed method

Among the different vectorial edge detection approaches available in literatures, the max gradient method [6, 7] and VOS [8] were considered in this thesis work as both are assumed to perform quite well. The implementation of the max gradient method was based on a recent version of the algorithm by J. Henriques which is freely available on the Matlab Central website. Figure 5.3 compares the performance of these two edge detection schemes with that of the proposed method in this thesis. Like the proposed edge detection technique, both max gradient as well as VOS performed quite satisfactorily in detecting edges on the synthetically generated color images. Significant performance issue was observed on the natural color scene composed of a bird sitting in the middle of a grass land. Both max gradient as well as VOS failed to detect the edges of the bird while, interestingly, the proposed scheme satisfactorily did its purpose.

Figure 5.4 repeats funds images presented in Figure 5.2 to compare the performance of the vectorial techniques. In this case VOS failed its test unable to detect objects of interest while the max gradient operator as well as the proposed technique performed quite well with the proposed technique giving less noise and much stronger edges than the max gradient operator.

The experimental results observed above clearly demonstrate the superior performance of the proposed trinion based color edge detection scheme explaining the importance of holistic approaches that make use of the systematic manipulation of the inter-correlation information that is embedded among the different monochromatic color channels/bands.

Figure 5.5, again demonstrates its applications in clearly contours the different features of fundus images collected from St. Paul's Hospital Millennium Medical College Ophthalmology Department. Here exudates, blood vessels have been identified clearly which could be used for effective image analysis techniques.

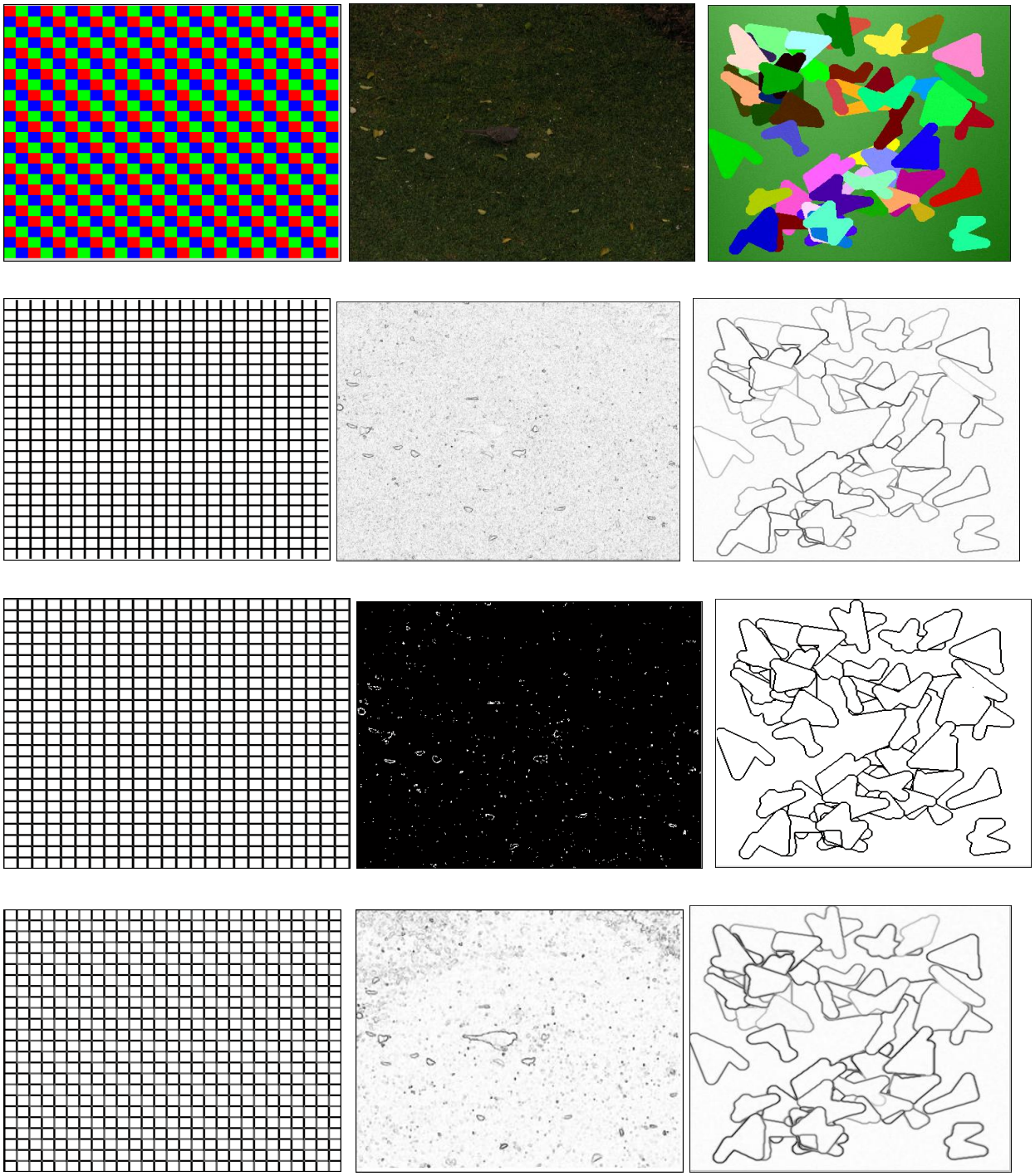


Figure 5.3 Original images (row 1), Vectorial max gradient approach (row 2), VOS edge map (row 3) and proposed technique (row 4)

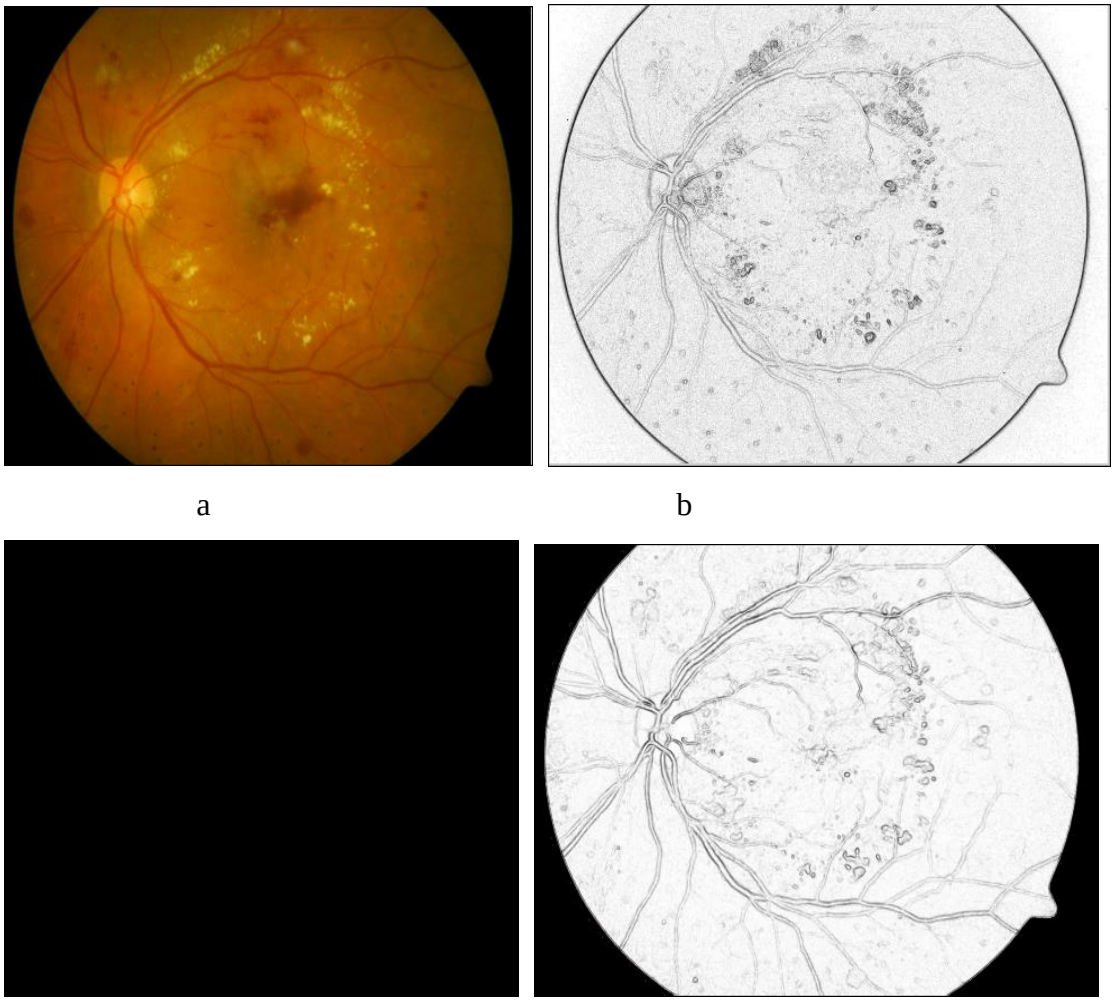


Figure 5.4 Original images (a), Vectorial max gradient approach (b), VOS edge map (c) and proposed technique (d)

Figure 5.5 Original images (a), Edge maps of the different features using the proposed method (d) [courtesy of St. Paul's Hospital Millennium Medical College Ophthalmology Department].

5.3. Quantitative Evaluation

Another field of study in edge detection which has not been given enough consideration is the performance evaluation of edge detectors [9]. It is natural to note that quantitative comparison of detected edge maps requires ground truth images. The main goal for this section is to use a metric that would evaluate any edge detector against a test set of images and generate numerical score for that edge detector. This way a new edge detector could quickly be evaluated, and a user could easily determine how similar an edge detector performs when compared to the human visual system. It assumes three stages: generate and collect a test set of images having ground truth, implement edge detectors and finally use a metric to numerically evaluate their performance against each other [9, 10].

However, manually constructing ground truth for real intensity images is problematic and even the definition of an edge is debatable. The difficulties involved in obtaining ground truth for real images are so great that researchers simply do not conduct quantitative evaluations of edge detectors using real images [11]. Evaluating the performance of edge detectors is an active research area by itself. In terms of methodology, it might be categorized as objective and subjective techniques. The first one uses machines to automatically judge their performance where the later uses human observers [12].

In order to be able to use a metric for edge detector effectiveness, it is necessary to have a test images for which the edges are known. Here we consider three cases: synthetically generated color images, pseudo-random and images taken from known databases in which case all have ground truths. Figure 5.6 presents the 288 by 288 synthetic image (presented in Figure 5.3 above) for comparison purposes. Figure 5.7 presents a pseudo-random image while Figure 5.8 presents an example test image taken from Berkley image segmentation and contour detection database. In all cases, the ground truth information is presented and compared against edge detection results by the proposed method in this thesis work. Note that the pseudo random image generator used in this thesis work creates images which are random within a certain set of bounds while these bounds are user defined but generally an overall effort was made to make the random images as natural as possible similar to the case made in previous studies reported in [8, 10].

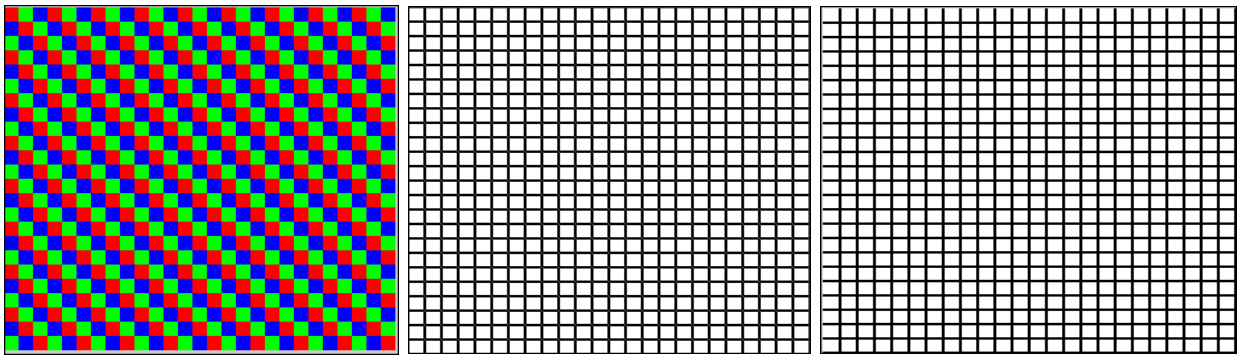


Figure 5.6 288 by 288 Matlab generated synthetic color image (left), ground truth (middle) and edge detection result by the proposed edge detection algorithm (right).

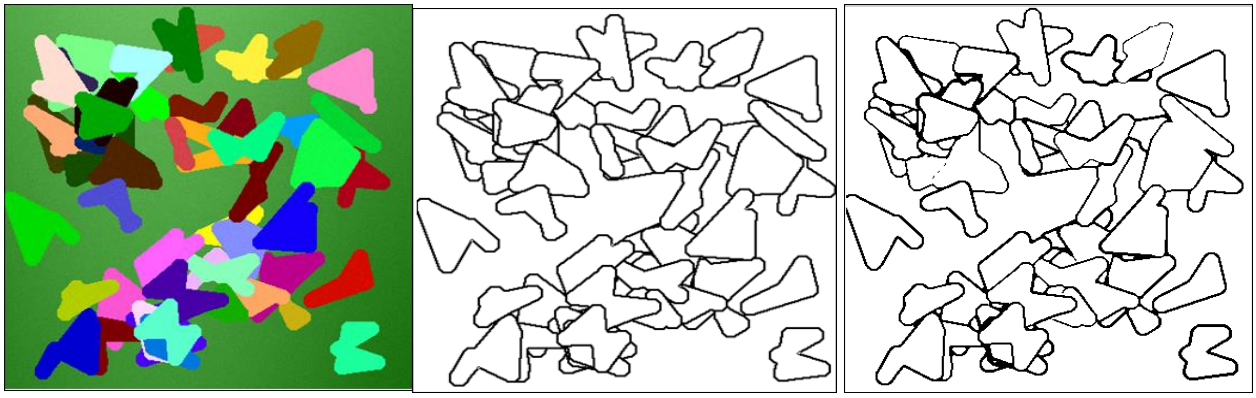


Figure 5.7 Pseudo-random generated image (left), Ground truth (middle) and Edge detection results by the proposed edge detection algorithm (right).

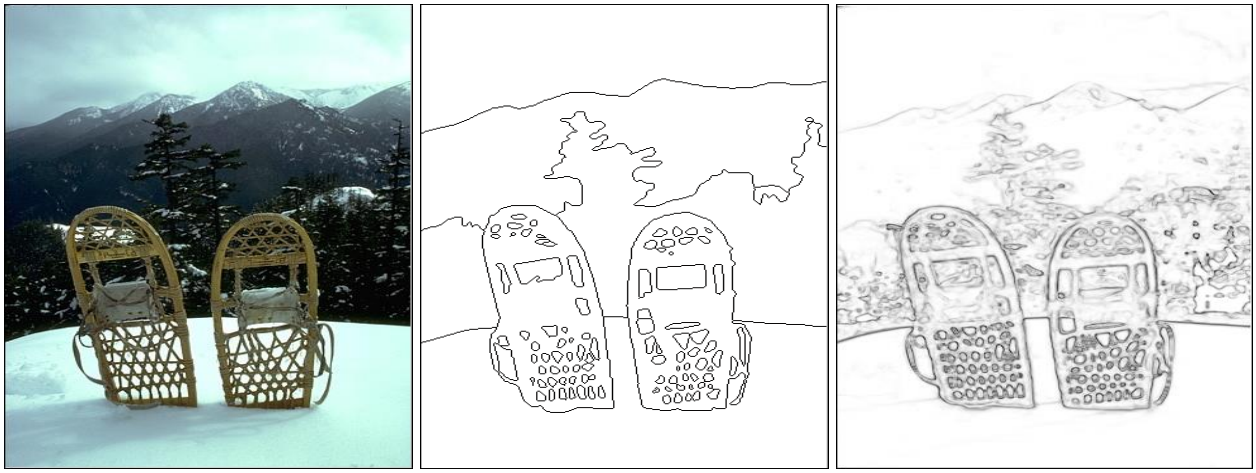


Figure 5.8 Image taken from Berkley image segmentation and contour detection database (left), ground truth (middle) and edge detection result by the proposed edge detection algorithm (right).

In order to quantify the performance of the different edge detection schemes, the metric used in this thesis study was Figure of Merit (FOM). FOM is sensitive to the different expected errors. The FOM, introduced by Pratt, is maximum when the edge map is perfect when compared against the ground truth and decreases as the error in the edge map increases [9, 11, 13]. FOM has two advantages; it weights the different errors according to their importance and it allows each edge detector to be tuned to its best capabilities, which guarantees a fair comparison. For edge maps taken from the Berkley segmentation and contour detection database and the pseudo random color images, it is obvious that further thresholding is needed to make binary edge maps. For that reason, an adaptive Otsu thresholding scheme has been used in this thesis work [18].

Generally, there are three possible errors while detecting edges in color image and generating the resulting edge map. These three major types of errors are missing of valid edge point, failure to localize edge points and classification of noise pulses as edge points [11]. Examples of these errors are demonstrated in Figure 5.9.

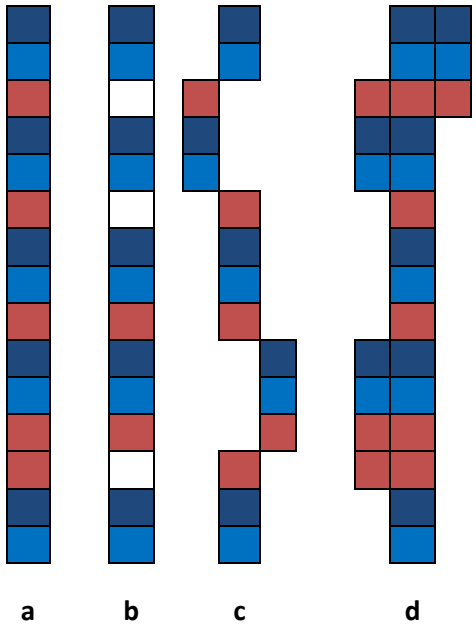


Figure 5.9 Edge error types. (a) ideal, (b) missed, (c) fragmented and (d) smeared edges [11].

The quality of the edge map generated by an edge detection operator is assessed by the FOM given in Eq. 5-1 [13, 14, 15, 16].

$$FOM = \frac{1}{max\,(I_I, I_D)} \sum_{k=1}^{I_D} \frac{1}{1 + \alpha(d_k^2)} \qquad 5\text{-}1$$

where $I_I \; and \; I_D$ represent the number of ideal and actual (detected) edge map points respectively; α is a scaling constant and d is the separation distance of an actual edge point normal to a line of ideal edge points. The rating factor is normalized so that FOM = 1 for a perfectly detected edge. The scaling factor α may be adjusted to penalize edges which are localized but offset from the true position. Normalization by the maximum of the actual and ideal number of edge points insures a penalty for smeared or fragmented edges [8, 10, 17]. Table 5-1 presents the average FOM computed for different edge detection schemes including the one proposed in this thesis considering the synthetic, pseudo random and data base generated test images.

Table 5-1 FOM (in percent) values computed for different edge detection schemes.

Type of edge detectors	Berkley image segmentation and contour detection database image of size 321 x 481 (154,401 pixels)	Matlab generated pseudorandom image of size 400 x 400 (160,000 pixels)	Matlab generated synthetic image of size 288 x 288 (82,944 pixels)
Prewitt	86.54	92.81	70.02
Sobel	86.54	92.85	70.02
Canny	88.08	94.64	70.02
Vector order statics	89.64	98.20	98.86
Max grad	90.56	97.22	99.21
Proposed technique	93.20	98.57	99.58

FOM values computed for different edge detection schemes are given in percent. Accordingly, the vectorial approaches (max grad, VOS and the proposed scheme) performed superior to the rest of the classical techniques. For the Berkley images, FOM of proposed method resulted in 93.20% followed by the max grad scheme which scored 90.20%. Out of the total 154, 401 pixels considered, the 3.26% difference maps to 5033 more mis-classified pixles by the max grad operator compared to the proposed scheme. For the Matlab generated images, the performance of max grad, VOS as well as the proposed scheme was reasonably close. As presented in the previous section, the performance of the proposed scheme when applied on natural color scenes with different degrees of difficulty as well as the acquired diabetic retinopathy fundus images is much superior to any of the rest of the techniques considered in this thesis work, be it classical or vectorial approaches. Quantitative evaluation was missing for those cases as no gold standards were available.

References

[1]. T. Kauppi, V. Kalesnykiene, J. Kamarainen, L. Lensu, I. Sorri, H. Uusitalo, H. Kalviainen, and J. Pieti, “DIARETDB0: Evaluation Database and Methodology for Diabetic Retinopathy,” pp. 1-17, 2008.

[2]. http://www.it.lut.fi/project/imageret/diaretdb0/

[3]. T. Kauppi, V. Kalesnykiene, J. Kamarainen, L. Lensu, I. Sorri, A. Raninen, R. Voutilainen, H. Uusitalo, H. Kalviainen and J. Pieti, “DIARETDB1 diabetic retinopathy database and evaluation protocol,” pp. 1-18, 2010.

[4]. http://www.it.lut.fi/project/imageret/diaretdb1/

[5]. https://www5.cs.fau.de/research/data/fundus-images/

[6]. Dawit Assefa and Ondrej Krejcar, “Novel Edge Detection Scheme in the Trinion Space for Use in Medical Images with Multiple Components,” Springer International Publishing Switzerland 2016, ICCCI 2016, Part II, LNAI 9876, pp. 231–241, 2016 DOI: 10.1007/978-3-319-45246-3 22.

[7]. https://www.mathworks.com/matlabcentral/fileexchange/28114-fast-edges-of-a-color-image-actual-color-not-converting-to-grayscale.

[8]. P. E. Trahanias and A. N. Venetsanopoulos, “Vector Order Statistics Operators as Color Edge Detectors,” IEEE Transactions on Systems, Man, And Cyberne,Tics-Part B: Cybernetics, 26(1), February 1996.

[9]. B. O. Sadiq, S.M. Sani and S. Garba, “Edge Detection: A Collection of Pixel based Approach for Colored Images,” International Journal of Computer Applications, 113(5), March 2015.

[10]. Amiya H., Nilabha C., Arindam K., Swastik P. and Soumajit P., “Edge Detection: A Statistical approach,” 3rd International Conference on Electronics Computer Technology (ICECT), 2(306), 2011.

[11]. Ikram Escandar Abdou, “Quantitative Methods of Edge Detection, Image Processing Institute University of Southern California Los Angeles,” Advanced Reasearch Projects Agency, California, July 1978.

[12]. Jiangyan Xu Weixing Wang and Linning Ye, “Rock fracture edge detection based on quaternion convolution by scale multiplication,” Optical Engineering 48(9), September 2009.

[13]. Phairoj Samutrak and Jeeraporn Werapun, “Efficient Color Edge Detection based on Synthetic Weighted multi-Structure Element Morphology,” Fifth International Conference on Graphic and Image Processing (ICGIP 2013), Proc. of SPIE, 9069(1), Dec 2013.

[14]. W. X. Wang and J.Y. Xu, “Edge Detection by Using Edge Density and Eleven Algorithm Comparisons in Three Types of Color Images,” SPIE-IS&T, 8292(44), Jan 2012.

[15]. Adrian N. Evans and Xin U. Liu, “A Morphological Gradient Approach to Color Edge Detection,” IEEE Transactions on Image Processing, 15(6), June 2006.

[16]. Ayaz Akram and Asad Ismail, “Comparison of Edge Detectors”, International Journal of Computer Science and Information Technology Research (IJCSITR), 1(1), pp. 16-24, October-December 2013.

[17]. Bunil Kumar Balabantaray et al., “A Quantitative Performance Analysis of Edge Detectors with Hybrid Edge Detector,” Journal of Computers, 12(2), March 2017 doi: 10.17706/jcp.12.2.165-173.

[18]. P. Roy and G. Dey et.al, “Adaptive Thresholding: A comparative study,” International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 978(1), pp. 1320-1324, July 2014, DOI: 10.1109/ICCICCT.2014.6993140.

Chapter Six

6. Conclusions and Future Works

6.1. Conclusions

Edge detection is a key tool for image segmentation, object detection and many other applications. Therefore, it is necessary to develop an effective edge detection system. In this regard, color edges are more informative compared to their gray scale counterparts. The traditional way of analyzing color edges requires separation of the different color channels/bands. By doing so, one will lose the inter-correlation information embedded within the color channels. A holistic approach should offer more commendable results. In that regard, the current thesis work proposed a scheme that makes use of trinions and their Fourier transforms to represent colors holistically and extract clever features for use in color edge quantification. Based on the comparison made against the classical edge detection technique (such as Prewitt, Sobel, LoG and Canny) and other vectorial approaches (including VOS and Max Grad), the proposed scheme was found to be significantly superior.

Many of the edge detection techniques discussed in the literature are very sensitive to noise. Furthermore, when it comes to colors, these methods take only intensity/brightness information into account. The more holistic approaches consider color pixels as one entity/object. Such holistic approaches are assumed to be more accurate and effective in many areas of color image processing other than edge detection [1, 2].

Various features were extracted from the color images in the trinion space to check their ability in effectively quantifying color edges. FOM was used to quantify their performance. Different synthetic as well as natural color scenes were considered while developing and testing of the proposed edge detection algorithm. While a medical application was shown on color image data acquired from retinal images of patients diagnosed with Diabetic Retinopathy (DR). In all cases considered, the proposed scheme outperformed all other edge detection techniques considered. Generally, the vectorial approaches performed superior to the classical serial approaches justifying why holistic approaches are preferred. The entire algorithm was implemented in a Matlab platform, on a core i3 Toshiba laptop computer, and it takes about 25 seconds for the

algorithm to detect and display the final color edge map for a typical color image of size 256 x 256. Such computational speed is acceptable while no effort was invested in the current work to improve the computational efficiency. Note that the discrete trinion Fourier transform was implemented by a repeated application of the fast Fourier transform (FFT).

6.2. Future Fork

There are still rooms to improve the performance of this edge detection scheme particularly as the color image under consideration gets more complex. After computing the windowed Fourier transforms, ways to extract even more powerful features than the once used in this study (second order variance, standard deviation, etc.) can be beneficial. This for example can mean searching for statistical features which are less sensitive to subtle changes within the color image such as noise. Testing for different color spaces other than the ones considered in this thesis work could also be interesting. The use of HPC (high performance computers) could be needed if the algorithm has to run in real time. Resolving these and similar other issues require further investigations.

References

[1]. Dawit Assefa and Ondrej Krejcar, “Novel Edge Detection Scheme in the Trinion Space for Use in Medical Images with Multiple Components,” Springer International Publishing Switzerland 2016, ICCCI 2016, Part II, LNAI 9876, pp. 231–241, 2016 DOI: 10.1007/978-3-319-45246-3 22.

[2]. D. Assefa, L. Mansinha, K. F. Tiampo, H. Rasmussen, and K. Abdella, “The trinion Fourier transform of color images,” Sig Proc., 91(8), pp. 1887-1900, 2011.