Medical Image Segmentation Using Spectral Clustering Based on Hypercomplex Analysis

By

Samuel Getaneh Bayih

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

Center of Biomedical Engineering

Addis Ababa Institute of Technology

Addis Ababa University

Advisors: Dawit Assefa Haile (PhD)

Birhanu Assefa Belay (MSc)

Addis Ababa, Ethiopia, October 2016

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 Samuel Getaneh Bayih entitled: *Medical Images Segmentation Using Spectral Clustering Based on Hypercomplex Analysis* submitted in partial fulfillment of the requirements for the degree of Master of Science in Biomedical Engineering (Bioinstrumentation and Imaging) 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_____

Advisor_____Signature_____Date_____

_____

Chief of Department or Graduate program coordinator

Abstract

Images are destined to be segmented, either with our visual system (as we do it every day) or algorithmically using computing machines. However, as critical as medical images, image segmentation has to be consistent which we do not usually have. Therefore, to bring consistency, physicians need some kind of consistent clues about medical images they are analyzing. Particularly, when the images considered are colors with multiple channels/bands, developing such a consistent segmentation tool could take more effort and that often requires a rigorous mathematical computation and algorithm free from human cognition. To create such algorithm for color medical image segmentation, primarily the images have to be considered as collection of pixels that have to be represented properly and holistically with no separation of color components. Secondly, the relationships of pixels have to be defined holistically. Finally, based on these defined relationships, the pixels have to be grouped and segmented. Hence, in order to perform holistic segmentation, quaternion based representation of color pixels and quaternion based spectral clustering technique has been proposed in this thesis. Test results have shown that the proposed scheme can use and also can be used in machine learning applications of image segmentation and pattern recognition.

Keywords: Quaternion Rotation, Spectral Clustering, Image Processing, Color Distance Measurement

Acknowledgement

It is a pleasure to thank the many people who made this thesis possible. First and foremost I would like to express my special appreciation and thanks to my advisor Dr. Dawit Assefa Haile (PhD.), you have been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to see things in different directions. I would also like to thank my co-advisor Birhanu Assefa Belay (MSc.) your efforts have been supportive since the day I began working on the thesis. I am also grateful to Dr. Masreshaw Demelash (PhD.), PG coordinator, for all his help and encouragement. A special thanks to my sister and brother-in-law, for their limitless support, love and patience. I would also like to thank the examiners, Dr. Frehiwot and DR. Solomon for their highly valuable comments. Finally I am grateful to all my friends who incented me to strive towards my goal.

Table of Contents

List of Figures

List of Tables

Chapter One

1 Introduction

1.1 Background

Basically, medical imaging is about acquiring data that describe the static and dynamic properties of the human[1] body. However, because of the complex nature of the human body and the huge data generated by it, researchers, clinicians and engineers are always asking how to acquire, process and display such data so that it can be used effectively for the development of useful diagnostic and therapeutic procedures. Since humans heavily rely on sight than any other perceptual skills to understand their environment, presenting a medical data as an image (static or dynamic) is known to be the better way for understanding it [1].

Medical imaging is a process that covers the interaction of electromagnetic waves, acoustic waves and the likes with biological structures, and the designing of systems to extract clinically important information which is then displayed as an image. Medical images reveal the characteristics of the human body such as its transmissivity, opacity, emissivity, reflectivity, conductivity, magnetizability, and their rate of change with time [1-4]. For instance, images created by X-rays transmitted through a region of the body reveal intrinsic properties of the region such as physical density (gm/cm^3), effective atomic number (Z_{eff}) and electron density ($electrons/cm^3$). Nuclear medicine images, like PET (Positron Emission Tomography) and SPECT (Single-Photon Emission Computed Tomography), reveal the spatial and temporal

[1]This document focuses on medical images of human subjects.

distribution of target-specific pharmaceuticals, for example, Fluorodeoxyglucose[2] (^{18}F-FDG) in the body. Data interpretation in nuclear medicine images depends on the application. The data can be interpreted as physiological processes like: glucose metabolism, blood volume, fluid flow and perfusion, tissue or organ uptake, receptor binding, and oxygen utilization. In ultrasonography (ultrasound imaging), images are produced by capturing energy reflected from the interfaces in the body that separate tissues with different acoustic impedances (which velocity of ultrasound wave depends on this impedance). In MRI (Magnetic Resonance Imaging), the magetizability of tissues or the relaxation property of tissues after magnetization is mainly influenced by the concentration, mobility, and chemical bonding of hydrogen atom in the tissues. EEG (Electroencephalography) at the surface of the skull can be used to identify the areas of higher neuroelectrical activity in the brain [1, 2]. We could go on and mention many more imaging protocols used in different clinics for various applications. Such techniques used for gathering information from the body are generally termed as imaging modalities. The different modalities allow us to capture medical images ranging from simple planar images to more sophisticated ones that have a capability of displaying temporal phenomena like functional Magnetic Resonance Imaging (fMRI) and four-dimensional medical images like a three-dimensional ultrasound image displayed in time domain.

The quality of medical images determines the accuracy of our understanding of them; however, they are often deteriorated by noise. During medical imaging, noises can be generated due to technical limitations of the device used, emission or detection of unnecessary electromagnetic waves, inability to detect the necessary electromagnetic waves, and variations in biological parameters of the subject. For instance, noise can be created due to oxidation of a dye because of

[2] The uptake of Fluorodeoxyglucose (^{18}F-FDG) is glucose uptake marker.

slide aging, movements of the subject in MRI images, etc. Such noises, if detected by the device, will be represented by specific pixel(s) in the medical images; pixels that were otherwise would represent the correct anatomical state or physiological activities of the subject. This will create a medical image that doesn‟t hold the whole correct data for analysis. Therefore, different mathematical pre-processing algorithms have to be applied on the noisy image to restore the „whole‟ correct data before it is analyzed. There have been a great number of pre-processing algorithms to remove different kinds of noise from medical images; even though their efficiency is dependent on the imaging modality and the type of noise degrading the image. Moreover, noise removal and enhancement algorithms are basically common to both medical and none medical images [9, 10, 11, 12, 13]. However, the discussion on preprocessing techniques is neither the scope nor the purpose of this document.

Medical images have irreplaceable role in detecting, treating and monitoring of diseases. For instance, the detection of tumors in their early stages determines the survival of the patient and the disease progression and this is highly dependent on medical images. Then their treatment procedure depends on, among other factors, tumor‟s size and location which can be determined usually from medical images. The effectiveness of the treatment is also assessed by analyzing medical images taken before, during or/and after the treatment [2-6].

The technological and biological science advancements have created high fidelity medical images. However, clinicians‟ interpretation of medical images depends on their professional experience, training level, and skill which could lead to inconsistent interpretation of medical images [14, 15] (as demonstrated in Figure 1-1 for example). Therefore, a consistent interpretation (i.e. detection, measurement and characterization) of medical images requires a consistent identification of the region of interest from the background. This means consistently

segmenting the entire image into multiple regions so that each region is considered a homogeneous region based on certain characteristics or features. This issue of observer variability is not the only one to be mentioned regarding the drawbacks of manual interpretation of medical images. It is also time consuming and often non-repetitive. In this regard, computer based image processing algorithms play a vital role in circumventing some of the shortcomings of the traditionally used manual techniques.

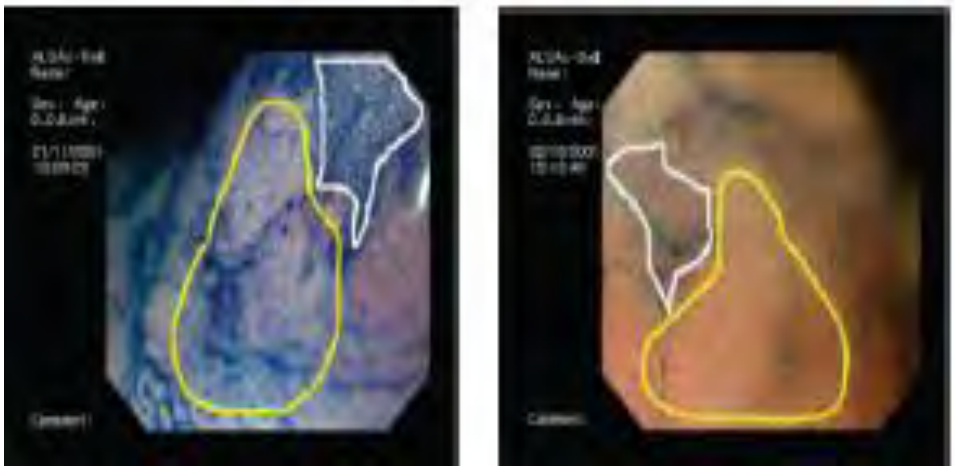


Figure 1-1 Manual segmentation of an endoscopy image to identify cancerous region by two clinicians showing inconsistent annotations (curtsy of A. Sousa et. al IEEE, 2009)

In medical imaging, segmentation is often a means to an end that has applications in tumor detection, lesion quantification, surgery simulation and planning, treatment evaluation, detection of micro-calcifications on mammograms, etc. Different computer based segmentation techniques have been developed to use for segmenting medical images into contiguous regions; however, there is no a standard medical image segmentation technique that works for all modalities [6 - 8].

1.2 Statement of the Problem

Even though the recent advancements in technology and biological sciences resulted in acquiring high fidelity medical images, identifying a target from the background has been a challenge. For example, the correct identification of cancerous tissues from medical images is a crucial step for radiation therapy planning. The efficiency of the therapy is highly dependent on the accuracy of the delineation, which requires the correct identification of the target from the background. However, the identification and delineation of cancerous tissues is highly dependent on profession specialty, level of training and personal bias, among other factors [14 - 17].

Many studies showed that, personal factors influence delineation of tumors during radiation therapy planning. For example, less experienced physicians tend to delineate larger volumes than the experienced ones in their field [14]. Other studies [16, 17] showed that health professionals selected from different specialties such as diagnostic radiologists (DR), radiation oncologists (RO) and medical physicists (MP) delineated different sizes of cancerous tissue from the same medical image. The variability was significant enough to be measured in centimeters. To offset such inconsistencies, many computer-based medical image segmentation techniques have been developed in various literatures.

One segmentation technique, for example, uses the intensity level of a pixel to decide whether it is an object or a background. Different algorithms are used to determine the threshold value(s) such that the intensity of a pixel will be considered as that of a background or not [6]. However, such a thresholding technique becomes inefficient if the number of different regions of interest (objects) in an image increases, contrast difference between the background and the object is

small, the image is noisy, or if the background intensity varies greatly. There are adaptive or local thresholding techniques but those are often computationally expensive [2, 6].

Another segmentation technique uses spatial filtering to calculate the first-order or second-order gradient information of the image so that the edges can be determined. Different operators of gradient calculation have been developed in the literature: Canny, Robert, Sobel, Laplacian, and others [3]. After calculating the gradient, to segment the image, the edge linking algorithm is applied using the calculated gradient information. The edge linking stage is usually very tedious because of the high probability of irregular gradient information and noise.

Another common segmentation technique defines similarity criteria so that two pixels or two regions or a region with pixel can be merged or not, region growing based. In another way, it defines dissimilarity criteria so that a region can be divided in to smaller „homogenous" regions, region splitting based [2, 3, 4]. This technique usually requires input information, called seed, from the operator; however, with different seeds, it results mostly different outputs.

As mentioned above, different techniques have been developed for generating a metric system for similarity or difference measures of pixels. Even though it works for gray scale images, most of the segmentation techniques be it pixel based, region based or edge based use scalar value representation of a pixel during processing, even a color pixel. However, using a non-vectored value to represent and process a color pixel will result in undermining the effect of inter-correlation of color components.

One vectorial approach, that has been proposed in the literature and has gained important attention recently in the world of color image processing uses a Quaternion number system [18] to represent a color pixel together with its multi-dimensional algebra. Quaternions allow holistic

analysis of color images keeping the inter-correlation of color components intact [19]. Therefore, this thesis will present automated, holistic image segmentation technique based on quaternions for representing color pixels and its related higher dimensional algebra for generating a metric system.

1.3 Objectives

1.3.1 **General objective**

- To develop an alternative segmentation technique that can segment both color as well as gray scale medical images.

1.3.2 **Specific objectives**

- To develop a scheme for segmenting color images based on a vectorial approach that takes into account the inter-correlation among color channels/bands.
- To show the effectiveness of the developed clustering technique in segmentation of synthetic as well as natural color scenes.
- To show the effectiveness of the developed clustering technique in segmentation of certain color medical images (gray scale images are special kinds of color images with equal color bands).

1.4 Significance of the Study

Personal factors are some of the reasons for high inconsistency in medical image segmentation or delineation. Such inconsistencies could somehow be reduced by giving trainings, enhancing professional collaborations and through experience. Due to financial and qualified manpower shortages, however, such measures often prove to be difficult. Therefore, automated medical image segmentation technique should augment the decision of the health professionals to be

relatively consistent. Moreover, this segmentation technique could be applied for both gray scale as well as color images and can be used to develop an effective machine learning technique.

1.5 Organization of the Thesis

This thesis comprises six chapters; the first chapter covers the background of the thesis. Chapter two discusses Quaternions and their basic properties. Then it discusses the most commonly used gray scale and color spaces and ways of representing color pixels in the Quaternion space. Chapter three discusses the fundamental concepts behind color image segmentation and the different classifications of segmentation techniques. It also summarizes the importance of medical image segmentation in general.

Chapter four discusses the core of this thesis work: the proposed segmentation technique which is Quaternion Based Medical Image Segmentation Technique. Details on how color pixels are represented holistically as a quaternion number, how similarity of color image pixels are measured, weighted and how they are clustered are included in the discussion. The chapter also incorporates brief discussions on the data sets that are used for testing the segmentation technique. In Chapter five, demonstrating results of the proposed scheme applied on both synthetic as well as medical images are presented and discussed in detail. Based on the results, Chapter six presents important conclusions and also points out possible future works. The appendix of this document presents some examples that can help to understand the difficulty of a manual determination of weighting function.

Reference

[1] William R. H, E. R. Ritenour, “Medical Imaging Physics”, John Willy and Sons, Inc. Publication, 2002

[2] Atam P. D., H. K. Huang, Dae-Shik K., ” Principles and Advanced Methods in Medical Imaging and Image analysis”, World Scientific Publishing Co. Pte. Ltd., 2008

[3] Atam P. D., “Medical Image Analysis”, Institute of Electrical and Electronics Engineers, Inc., 2011

[4] Klaus D. T., “Guide to Medical Image Analysis: Methods and Algorithms”, Springer-Verlag London Limited, 2012

[5] G. Dougherty, “Medical Image Processing Techniques and Applications”, Springer, 2011

[6] I. N. Bankman, “Handbook of Medical Imaging, Processing and Analysis”, Academic Press, 2009

[7] E. R. Davis, “Computer and Machine Vision: Theory, Algorithms and Practices”, Elsevier Inc., 2012

[8] M. Sonka, V. Hlavac, R. Boyle, “Image Processing, Analysis and Machine Vision”, 3rd Edition, Thompson Learning, 2008

[9] T. M. Deserno, “Biomedical Image Processing”, Springer-Verlag Berlin Heidelberg, 2011

[10] H. Lu, X. Li, Z. Liang, I. T. Hsiao, “Analytical Noise Treatment for Low-dose CT Projection data by penalized weighted least-squares smoothing in K-L domain”, proc. SPIE, Medical Imaging 2002, Vol. 4682, pp. 146

[11] J. Sijbers, “Signal and Noise Estimation from Magnetic Resonance Images”, Ph.D. thesis, University of Antwerp, Belgium, 1999

[12] R. C. Gonzalez, R. E. Woods, “Digital Image Processing”, 3rd Edition, Pearson International Edition, 2008

[13] R. C. Gonzalez, R. E. Woods, S. L. Eddins, “Digital Image Processing using Matlab™ ”, Prentice Hall, 2004

[14] C. F. Nieh, “Tumor delineation: the weakest link in the search for accuracy in radiotherapy”, Journal of medical physics, 2008, 33(4):136 - 140

[15] A. Sousa, M. Dinis-Ribero, M. Areina, M. Coimbra, “Identifying Cancer Regions in Vital-Stained Magnification Endoscopy Images Using Adapted Color Histograms”, IEEE, 2009

[16] D. H. Wu, N. A. Mayr, Y. Karatas, R. Karatas, M. Adli, S. M. Edwards, et. al, “Interobserver Variation in Cervical Cancer Tumor Delineation form Image-based Radiotherapy Planning Among and Within Different Specialties”, Journal of Applied Clinical Medical Physics, Vol. 6, No. 4, 2005

[17] E. Weiss, C. F. Hess, “The Impact of Gross Tumor Volume (GTV) and Clinical Target Volume (CTV) Definition on the Total Accuracy in Radiology”, Strahlenther Onkol, 2003, 179: 21-30

[18] William Rowan Hamilton, “Lectures on Quaternions,” Dublin, 1853

[19] D. Assefa, L. Mansinha, K. F. Tiampo, H. Rasmussen, and K. Abdella, “Local Quaternion Fourier Transform and Color Image Texture Analysis”, Sig Proc., 90(6), pp. 1825-1835, 2010.

Chapter Two

2 Quaternions and Color Images

2.1 Quaternions

Quaternions form a non-commutative number system which is an extension of the 2D complex number system and was first described by Hamilton in 1843. The four-dimensional space, $\boldsymbol{Q}$ of quaternions is described by one real axis with three orthogonal axes, spanned by vectors i, j, k, called principal imaginaries [1-3].

$$\boldsymbol{Q = w + ix + jy + kz} \qquad \textit{2-1}$$

The principal imaginaries obey the Hamilton‟s rules, shown in equation 2-2.

$$\boldsymbol{i^2 = j^2 = k^2 = ijk = -1}, \qquad \textit{2-2}$$

$$\boldsymbol{ij = k, jk = i, ki = j \; and \; ji = -k, kj = -i, ik = -j}$$

Equation 2-2 shows that quaternion multiplication creates rotation. For instance, the multiplication of $\boldsymbol{k}$ to the left of $\boldsymbol{i}$ rotates $\boldsymbol{i}$ into $\boldsymbol{j}$ and multiplication of $\boldsymbol{k}$ to the left of $\boldsymbol{j}$ rotates $\boldsymbol{j}$ into $\boldsymbol{-i}$.

Modulus of a quaternion number is calculated as shown in equation 2-3 and it is called unit quaternion when it equals one.

$$\boldsymbol{|Q| = \sqrt{w^2 + x^2 + y^2 + z^2}} \qquad \textit{2-3}$$

A quaternion with real component equals zero is a pure quaternion, $\boldsymbol{v}$ as shown in equation 2-4.

$$\boldsymbol{v = ix + jy + kz} \qquad \textit{2-4}$$

2.1.1 **Basic properties of Quaternions**

Quaternions are different from both real and complex numbers; therefore, they have some distinct properties [2, 3]:

a. Quaternion multiplication is non-commutative; therefore, it has different exponential property than both real and complex number systems. We could for example look at the following equality that holds in the natural complex space but not with quaternions.

$$\boldsymbol{e^{ix+jy} \neq e^{ix}e^{jy}} \qquad \textit{2-5}$$

b. A Quaternion can be presented as a combination of a scalar and a vector as:

$$\boldsymbol{Q = S(q) + V(q)} \qquad \textit{2-6}$$

$$\textit{where: } \boldsymbol{S(q) = w} \textit{ and } \boldsymbol{V(q) = ix + jy + kz}$$

c. Quaternions can be written in polar form as follows

$$\boldsymbol{Q = |Q|e^{\mu\theta}} \qquad \textit{2-7}$$

where: $\mathbf{|Q|}$ is the modulus of the quaternion $\mathbf{|Q| = \sqrt{w^2 + x^2 + y^2 + z^2}}$,

$\boldsymbol{\mu}$ is Eigen axis of the quaternion $\boldsymbol{\mu = \frac{V(q)}{|V(q)|}}$

$\boldsymbol{\theta}$ is Eigen angle of the quaternion $\boldsymbol{\theta}$= $\tan^{-1}\frac{|V(q)|}{S(q)}$, $0 \leq \theta < \pi$

The Eigen axis, $\boldsymbol{\mu}$, of a quaternion is a *pure* and *unit* quaternion with $\mu^2 = -1$.

d. The quaternion field (4D) can be taken as combination of two complex fields (2D), i.e. $\boldsymbol{Q = w + ix + jy + kz}$ can be expressed as $\mathbf{Q = C_1 + C_2 j}$ where $\mathbf{C_1 = w + ix}$ and $\mathbf{C_2 = y + iz}$ which shows that **Q** is a two dimensional complex space or $\mathbf{C^2_i}$.

2.1.2 **Quaternion Rotation**

The rotation of a point in a space can be represented effectively with unit quaternion, **q**, where **q** = {q: |q|=1}. A rotation of a point with angle θ about an axis **v**, which is a unit and pure quaternion, can be represented as [4].

$$\mathbf{q} = \cos\frac{1}{2}\boldsymbol{\theta} + \mathbf{v}\sin\frac{1}{2}\boldsymbol{\theta} \qquad \textit{2-8}$$

The form of Equation 2-8 can be taken as a complex number, **q** = a + **v**b, though the imaginary part of the number here is **v**. This new imaginary axis **v** is a unit and pure quaternion. Since Euler‟s and De Moivre‟s hold in quaternion space equation 2-8 can be written as in exponential form [3, 4]:

$$\mathbf{q} = e^{\frac{1}{2}\boldsymbol{\theta}\mathbf{V}} \qquad \textit{2-9}$$

Equation 2-9 is another description of the rotation of a point through an angle $\boldsymbol{\theta}$ about the axis **v**.

The unit quaternion **q** in equation 2-9 can be applied to any pure vector, **p**, so that this **p** can be rotated. The rotation of **p** by angle $\boldsymbol{\theta}$ about the axis **v** can be computed as shown below [2, 4, 5].

$$\mathrm{QR}(\mathbf{p}) = \mathbf{qpq}^{-1} = \mathbf{qpq}^{*} \qquad \textit{2-10}$$

The above holds since, $\mathbf{q}^{-1}$ is equal to $\mathbf{q}^{*}$ (conjugate of **q**) for any *unit* quaternion **q** [2 - 4]. The quaternion rotation of any unit quaternion **p** by angle $\boldsymbol{\theta}$ about an axis **v**, which is given by the unit and pure quaternion **q** in equation 2-9, is as shown in equation 2-10. By selecting the rotation angle $\boldsymbol{\theta}$ and the rotation axis **v**, it is possible to rotate a vector in an angle $\boldsymbol{\theta}$ about an given axis **v**.

2.2 Color Image Representation

Electromagnetic wave is a physical phenomenon that can be measured; however, our perception of it as a color may be well measured, quantified and communicated but not precisely. This makes representation and quantification of colors a difficult task. Therefore, image processing algorithms require better precision, especially images holding medical information [6 - 8].

Trichromatic nature of the human visual system compels us to represent color images in three coordinate spaces so that three stimuli values can be represented. However, the three coordinate color space may not be completely suitable for performing image calculations or encoding color component data, since there are infinite number of possible values that do not correspond to any color phenomena. Some of the reasons for these discrepancies in color space values and their corresponding color representation are: limitations in physical realization of imaging and displaying technologies, our encoding capability for displaying and transmission of color data, consideration of linear relationships between different color components, and intuitive color specifications. Due to these limitations and requirements, there is no a single color space which suits for all sorts of color analysis [8]. In this chapter, it is found necessary to discuss only color spaces that are used in the research; therefore, in the next subsections of this chapter only Gray, RGB and HSL spaces are discussed. For further discussions on the topic, readers may consult previous works in the literature [7, 8, 11].

2.2.1 **Gray Space**

It is the simplest space for image representation. Images represented in gray space are those their pixel is black, white or different shades of gray. Each pixel has a single value that represents a shade which is the level of blackness or whiteness with no other color component in it.

Therefore, gray scale images have less information than color images. Gray scale is calibrated between pure white with gray value 1 (or 255), pure black with gray value 0, and in between values 0 and 1 (or 255) there are 254 different gray level shades. These values are enough to represent different gray levels that human eyes can differentiate [3].

2.2.2 **RGB Space**

The RGB (an acronym of Red, Green, and Blue) space is derived to represent color images with their primary colors: red, green, and blue. The idea of the RGB color space is to stick to the principle of human vision and represent colors as a simple sum of any quantities from 0 to 1 (or from 0 to 255) of the primary colors. As such it can be represented as a simple cube where three of the vertices along the axes represent the primary colors.

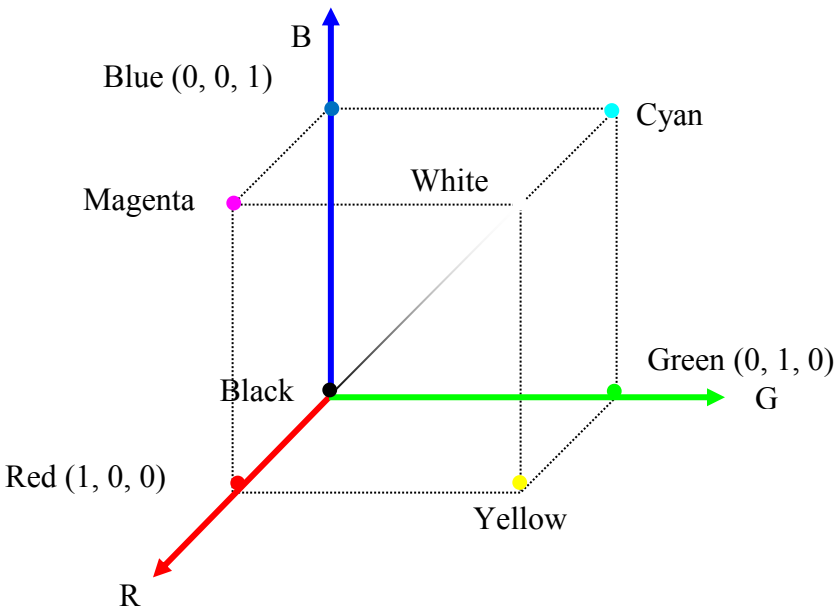


Figure 2-1 The RGB color Space

As it can be seen in Figure 2-1, any point inside or on the surface of the cube represents a color with the amount of primary color components with corresponding values on the respective axes.

The vertices (0, 0, 0) and (1, 1, 1) are special as they correspond to white (when the three primary colors are mixed up together in full amount) and black (absence of any of the three primary colors). The diagonal of the cube between the vertex (0, 0, 0) and vertex (1, 1, 1) represents the gray line (scale). A point on the gray line means its components (R, G, and B) have equal values, i.e. a gray scale image can be represented on the RGB space with all its primary color values equal.

The RGB representation only describes the chromatic values of the primary colors and their additive combinations; however, this is not the only visual effect any color has. There are more meaningful color features that have to be considered, like: how green is a color? Is it bright green or dark green? Two green colors can have identical primary chromatic content but difference in their intensity or luminance. Such features cannot be represented with their RGB values; they require a different color space one of them being the HSI color space [3].

2.2.3 **HSI Space**

The HSI (Hue, Saturation and Intensity) color space is believed to be the closest of color representations to human visual perception.

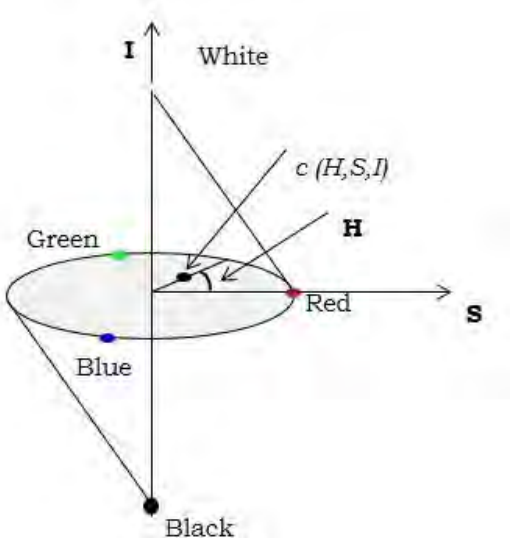


Figure 2-2 The HSI color space

Figure 2-2 shows that if a color ***c*** *(R, G, B)* is given, it can be represented in HSI space. The **H** component represents the dominant color in **c** (i.e. the hue) and it is represented in degrees from the **S** axis. Red is the reference color for measuring hue and it is on the intersection point of the unit circle with the S axis. Red is assumed to be reference color for measuring hue because it is at 0^o or 360^o from the saturation axis. The other primary colors green and blue are at 120^o and 240^o on the unit circle, counter clockwise from the saturation axis.

Numerically, RGB values can be converted to HSI by using the formulae described below:

$$\mathbf{H} = \begin{cases} \boldsymbol{\delta} & \boldsymbol{if\ B \leq G} \\ \boldsymbol{360^o - \delta\ \ if\ B > G} \end{cases} \qquad \textit{2-11}$$

where:

$$\boldsymbol{\delta = \cos^{-1}\left[\frac{(R-G)+(R-B)}{2\sqrt{(R-G)^2+(R-B)(G-B)}}\right]} \qquad \textit{2-12}$$

The **S** value of **c** shows the saturation with a specific color component (not only the primary component). Therefore, saturation value is a measurement of color purity. This parameter is dependent on the number of constituent color components that contribute for its perception. If

the number of wavelengths contributing in the perception of **c** is higher, then the purity of **c** decreases, which means the saturation value decreases. For the extreme values: S = 1 shows the color is pure and S = 0 shows it is achromatic.

S can be computed from RGB values as:

$$\boldsymbol{S = 1 - 3\frac{min(R,G,B)}{R+G+B}} \qquad \textit{2-13}$$

Intensity **I** of **c** corresponds to the relative brightness. If the brightness is zero that means no color is bright enough to be visible, resulting in black color perception. If the intensity value is 1 that means all the color components have a maximum brightness, resulting in white color perception [7]. Intensity of a color can be computed from RGB values as:

$$\boldsymbol{I = 0.2989R + 0.5870G + 0.114B} \qquad \textit{2-14}$$

The conversion of color values from RGB space to HSI space is reversible, except some inaccuracies due to round off errors.

2.3 Image Representation in the Quaternion Space

As described earlier, most of the image segmentation techniques use different representations for color and gray scale images. The pixels are represented either in real or complex space. Such a representation may be adequate for gray scale pixels. But as discussed above, color images, which could be represented in different color spaces, come with three components (or four components, for example as in the CMYK (Cyan, Magenta, Yellow, Black) color space). The real or complex representation is still used to represent such color images. But that essentially requires separation of color components. For example, during color image segmentation, the primary color components of a pixel are taken separately and analyzed as achromatic or

monochromatic images. After that, they are recombined and interpreted as color images. Such analysis techniques assume the interrelationship of the primary color components as linear, although they are not [3, 6]. Furthermore, we might want to, for example, compute the Fourier transform of the colors without separating the color components. In such and other applications, real or complex representation of the color pixels plays a very limited role. This calls for a more holistic representation of the color pixels.

Different studies have been done previously to analyze color images holistically so that the interrelationships of primary color components are intact [9, 10]. Each color pixel can be represented as a point in the RGB space, which can easily be mapped to a quaternion and one common way of doing that is to use a pure quaternion with the three color components mapped to the imaginary components assuming the real part is zero. Therefore, the corresponding RGB value of a pixel at position $\boldsymbol{p(x, y)}$ can be represented in the quaternion space as:

$$\boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{R}(\boldsymbol{x}, \boldsymbol{y})\boldsymbol{i} + \boldsymbol{G}(\boldsymbol{x}, \boldsymbol{y})\boldsymbol{j} + \boldsymbol{B}(\boldsymbol{x}, \boldsymbol{y})\boldsymbol{k} \qquad \textit{2-15}$$

Equation 2-15 shows that a color pixel at $(\boldsymbol{x}, \boldsymbol{y})$ can be represented with a pure quaternion. The three axes of the RGB space correspond with the three imaginary components of the quaternion number. The same scheme can be used to map a gray scale pixel to a quaternion by setting each imaginary component of the quaternion assume the single gray value of the pixel. Any analysis that is performed on the pure quaternion number $\boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y})$ will be applied to all of its color components, without affecting the interrelationships of the red, green and blue color components [3, 6, 9, 10]. Same analogy works if the color is represented in any other color space such as HSI, i.e. we can analyze the HSI colors holistically without separating the chromaticity and the luminance components.

Based on the above color pixel representation scheme in the quaternion space, preceded by some background principles, the new clustering tool will be developed in the subsequent chapters for use in complete segmentation of color images based on similarity metrics computed locally over the given color image.

References

[1] William Rowan Hamilton, “Lectures on Quaternions,” Dublin, 1853

[2] John C. H., Gregory K. F and Louis H. K., “Visualizing Quaternion Rotations”, ACM Transactions on Graphics, Vol. 13, No. 3, July 1994, Pages 256-276.

[3] Dawit Assefa, “1D and 2D Localized Hartley Transforms, their parallel implementations and applications; color image analysis using Quaternions and Trinions”, PHD Dissertation, London, ON, Canada, 2007

[4]Ken Shoemaker, 1985, ”Animating Rotations with Quaternion Curves”, ACM SIGGRAPH Computer Graphics, Volume 19, Issue 3, Jul. 1985

[5]Xiang L., Lianghai J., Hong L., Zeng H., ”A Quaternion-based spectral clustering method for color image segmentation”, Proc. of SPIE, Vol.8003, 800303, 2013.

[6] Dawit Assefa, Lalu M., Kristy F. T., Henning R., Kenzu A., “The Trinion Fourier Transforms of Color Images”, Elsevier, Signal Processing 91 (2011), 1887 - 1900

[7] Andreas K., Mongi A., “Digital Color Image Processing”, John Willey & Sons, 2008

[8] Erik R., Erum A. K., Ahmet O. A., Gareth M. J., “Color Imaging: Fundamentals and Applications”, 2008, A K Peters, Ltd.

[9] S. J. Sangwine, "Fourier transforms of colour images using quaternion or hypercomplex numbers, "*Electron. Lett.* 32(21), 1979-1980 (1996)

[10] S. C. Pei and C. M. Cheng, "A novel block truncation coding of color images by using quaternion-moment preserving principle," IEEE Int. Symp. Circuits Syst. 2, 684-687 (1996)

[11] Konstantinos N. P., Anastasios N. V.," Color Image Processing and Applications", Springer-Verlag Berlin Heidelberg 2000.

Chapter Three

3 Image Segmentation Using Clustering

3.1 Image Segmentation

Image segmentation is the process of dividing an image into non-overlapping, contiguous image areas based on some homogeneity criteria. The segmented regions then will have a meaning as regions of interest which are called foreground (objects) and the rest are called background [1, 2]. Traditionally, image segmentation has been based on gray scale analysis in case of monochromatic images or based on individual color component analysis in case of colors. Segmentation of color images is considered superior to their gray scale counterparts. The problem of segmenting an image based on gray scale variations is that it contains only intensity variations of shades ranging from pure black to pure white. Moreover, segmentation becomes more difficult when the background and foreground have a small difference between their intensity values on the gray scale. Besides, human eyes can only differentiate only a few dozens of gray scale levels due to brightness adaptation but can differentiate thousands of different color shades and intensities [3, 4]. Therefore, it is a natural way to analyze images in a color space.

Color image segmentation describes the process of extracting regions satisfying uniformity criteria based on features derived from spectral color components of the image. Due to the multidimensional nature of color information and also to keep the intrinsic correlation between color components, different vectorial approaches have been used in the literature. One of such techniques that has shown great promises in various color image processing applications uses quaternions and the respective algebra, one of these applications being segmentation [1-4]. The

technique allows holistic representation of color pixels by treating each as one entity. However, "There is no theory of image segmentation. As a consequence, no single standard method of image segmentation has emerged. Rather, there are a collection of ad hoc methods that have received some degree of popularity" [5]. Quaternion based segmentation schemes have rather shown solid performances compared to most of the „ad-hoc" methods mentioned in the previous quote. There have been various methods followed in the literature for segmenting an image, but the basic formalisms and concepts are identical as discussed in the next section.

3.2 Fundamental Concepts of Color Image Segmentation

3.2.1 **Formalisms**

Mathematically, a digital image can be modeled as a function $\boldsymbol{I}(\boldsymbol{x},\boldsymbol{y})$ that maps a location $(\boldsymbol{x},\boldsymbol{y})$ in space to a pixel value $\boldsymbol{I}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{v}$. Pixels of „black and white" images represent black, white or gray in discrete values ranging from 0 to 255. This means $\boldsymbol{I}(\boldsymbol{x},\boldsymbol{y})$ can be represented with a single scalar value between 0 and 255. Currently, the relative ease in taking color images and the better information content of color imaging made color imaging the preferred mode of imaging. The pixels of color images $\boldsymbol{I}(\boldsymbol{x},\boldsymbol{y})$ represent the color components (usually Red, Green and Blue) for each space located on the image position $(\boldsymbol{x},\boldsymbol{y})$. But this pixel representation is composed of the three values of the component colors (say red, green and blue). This means the value of each color pixel can only be represented as a vector not a scalar. Each color component, sometimes called color channel, represented by integer or floating point numbers and in all cases the pixel $\boldsymbol{I}(\boldsymbol{x},\boldsymbol{y})$ will have a vector value as $(\boldsymbol{r},\boldsymbol{g},\boldsymbol{b})$ [2].

An Image $\boldsymbol{I}$ is a finite rectangular grid of $[\boldsymbol{0}\ldots\boldsymbol{M}]\boldsymbol{x}\,[\boldsymbol{0}\ldots\boldsymbol{N}] \subseteq \mathbb{N}^{\boldsymbol{2}}$ and can be represented as a function of discrete natural number values:$\mathbb{N} \rightarrow \boldsymbol{V}$. Pixels of gray-scale images can be

represented with values $\boldsymbol{V} = [\mathbf{0}, \ldots, \mathbf{255}] \subseteq \mathbb{N}$ and those of color images with values $\boldsymbol{V} = [\mathbf{0} \ldots \mathbf{255}]^{\mathbf{3}} \subseteq \mathbb{N}^{\mathbf{3}}$.

An image **I** is said to be completely segmented into a number $\boldsymbol{N_R}$ of regions $\boldsymbol{R_i}$ with $\boldsymbol{i} = \mathbf{1} \ldots \boldsymbol{N_R}$ if the regions exhibit the followings [2 - 4]:

- The union of all regions should give the entire image: $\cup_{\boldsymbol{i=1}}^{\boldsymbol{N_R}} \boldsymbol{R_i} = \boldsymbol{I}$;
- The regions should not overlap : $\boldsymbol{R_i} \cap \boldsymbol{R_j} = \emptyset \; \forall \boldsymbol{i} \neq \boldsymbol{j}$;
- Each segment $\boldsymbol{R_i}$ is a connected component i.e., the pixel locations $\boldsymbol{P} \in \boldsymbol{R_i}$ in a region $\boldsymbol{R_i}$ are connected;
- Pixels belonging to the same region have similar properties under certain uniformity criteria: $\forall \boldsymbol{i}$, when a certain uniformity criteria $\boldsymbol{\zeta}(\boldsymbol{R_i})$ is satisfied, $\boldsymbol{\zeta}(\boldsymbol{R_i}) = \boldsymbol{TRUE}$;
- Pixels belonging to different regions do not satisfy the same uniformity criteria simultaneously: $\forall \boldsymbol{i} \neq \boldsymbol{j}$ if $\boldsymbol{\zeta}(\boldsymbol{R_i}) = \boldsymbol{TRUE}$ then $\boldsymbol{\zeta}(\boldsymbol{R_j}) = \boldsymbol{FALSE}$ or vice versa.

The segmentation formalism states that the image is decomposed into a number $\boldsymbol{N_R}$ of regions $\boldsymbol{R_i}$, with $\boldsymbol{i} = \mathbf{1} \ldots \mathbf{N_R}$, which are disjoint nonempty sections of $\boldsymbol{I}$, as shown below in Figure 3-1.

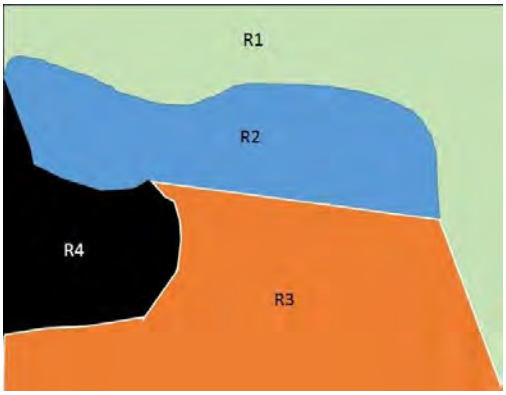


Figure 3-1 A theoretical example of complete image segmentation

A good segmentation scheme creates „homogeneous" component regions with respect to some defined homogeneity characteristics and the regions of the adjacent segments will have noticeably different values with respect to the characteristics on which they are uniform. The boundaries of each segment will be spatially accurate [2, 5]. Achieving all these is dependent on the choice of homogeneity criterion $\boldsymbol{\zeta}$ for the segmentation.

3.2.2 **Homogeneity**

In image segmentation, homogeneity is the visual content considered as inseparable whole. It is possible to define homogeneity criteria $\boldsymbol{\zeta}$ to check a segment $\boldsymbol{R_i}$ is either homogeneous or not. For example, binary homogeneity can be defined based on certain threshold, shown in equation 3-1 below.

$$\boldsymbol{\zeta(R_i)} = \begin{cases} \boldsymbol{TRUE}, & \boldsymbol{if}\ \ \forall \boldsymbol{P} \in \boldsymbol{R_i}: \left|\left|\boldsymbol{I(P)} - \boldsymbol{\mu(R_i)}\right|\right| \leq \emptyset \\ \boldsymbol{FALSE}, & \boldsymbol{Otherwise} \end{cases} \qquad 3\text{-}1$$

where: $\boldsymbol{\mu(R_i)} = \frac{\mathbf{1}}{||\boldsymbol{R}||}\sum_{\boldsymbol{P'}\in\boldsymbol{R}} \boldsymbol{I(P')}$ is the mean of pixel values in the region *i*, and $\emptyset$ is the threshold value.

Equation 3-1 defines region of a segment where no pixel differs from the mean of the pixel values inside the region by more than a threshold value $\emptyset$. For example, this equation measures the difference which requires a range that supports algebra of addition, subtraction, division and norm. Such mathematical measurements do not yield a unique segmentation because depending on the processing strategies and algorithms, different segments can be created from the same image [2, 3, 5]. As complexity, resolution, color texture variability, etc. of an image increase, image segmentation will require a more robust and enhanced homogeneity definition.

3.2.3 **Similarity Measures**

Image segmentation searches for variables that maximize the heterogeneity between segments and homogeneity inside the segments. But usually, variables are chosen from gray level, color spectrum distribution, and rarely from color texture. Several issues are considered in measuring similarity. The first issue that needs to be considered is pixel color and how to measure the distance between colors and then how to measure the color distribution so that textures can be quantified. These require the color space selection to compute similarity and then to compute features. Measuring colors of pixels and their differences in color coordinate is dependent on the application. Especially, when human perception is disregarded, the difference between colors can be

judged from a purely mathematical point of view; therefore, any distance matric between vectors can be used to express the similarity or the difference between colors [2, 3, 5].

3.2.3.1 **Color Distances**

As it is discussed in the previous chapter, color spaces are used to represent color components as vectors in three-dimensional space. However, the three-dimensional color produced by plotting three stimulus values (x, y, z) in a rectangular or cylindrical coordinates are not visually uniform. Equal distances in these spaces do not represent equally perceptible differences between color stimuli [1, 2]. Therefore, International Commission on Illumination (Commission Internationale de l"Edairage, abbreviated CIE) recommended lab and luv color spaces to compute color differences [1]. These color spaces are developed by CIE to linearize the difference of color perceptibility. After linearization, the „perceptual" color difference as it is observed by human eyes can be calculated from Euclidean distance Δe. For two color components (l_1, a_1, b_1) and (l_2, a_2, b_2) the difference Δe is

$$\Delta \boldsymbol{e} = \sqrt{(\boldsymbol{l_1} - \boldsymbol{l_2})^2 + (\boldsymbol{a_1} - \boldsymbol{a_2})^2 + (\boldsymbol{b_1} - \boldsymbol{b_2})^2} \quad 3\text{-}2$$

Because the human eye is more sensitive to hue differences than chromatic and lightness differences, the Euclidean distance $\Delta \boldsymbol{e}$ used to calculate color value differences falls short of its target. As it is shown in equation 3-2, $\Delta \boldsymbol{e}$ only uses chromatic values for calculating perceptual color differences. This led to continuous updates in the mathematical computation of $\Delta \boldsymbol{e}$, further detailed discussions can be found in [1, 3].

3.2.3.2 **General Purpose Distances**

In order to segment an image, it is required to quantify the homogeneity property and it is usually done by defining and calculating differences or distances. Then segmentation is done by minimizing these differences in the region of a segment. The distances commonly used for this purpose are discussed next.

The Minkowski distance of order p between two points $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ is defined as:

$$\boldsymbol{d}(\boldsymbol{X}, \boldsymbol{Y}) = \left[\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} |\boldsymbol{x_i} - \boldsymbol{y_i}|^{\boldsymbol{p}}\right]^{\frac{\boldsymbol{1}}{\boldsymbol{p}}} \quad 3\text{-}3$$

where $\boldsymbol{p}$ = 1 or 2. When $\boldsymbol{p}$ = 1 it is called Manhattan distance and when $\boldsymbol{p}$ = 2 the Minkowski distance becomes Euclidean distance. As $\boldsymbol{p} \to \infty$ it becomes Chebyshev distance shown below.

$$\lim_{\boldsymbol{p}\to\infty} \left[\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} |\boldsymbol{x_i} - \boldsymbol{y_i}|^{\boldsymbol{p}}\right]^{\frac{\boldsymbol{1}}{\boldsymbol{p}}} = \max_{\boldsymbol{i=1,\ldots,n}} |\boldsymbol{x_i} - \boldsymbol{y_i}| \quad 3\text{-}4$$

The difference measurements listed in the above equations (equation 3-2 to equation 3-4) and also various others [2] can be used to measure the difference between pixel values. Therefore,

segmentation of images can be performed by using these distances as a parameter for determining the segmentation level i.e. to what degree the segmentation is done.

3.3 Color Image Segmentation

Defining color spaces (refer back to chapter two) is the basic step in segmenting color images since it determines the performance of the segmentation [1 - 5]. But, at the same time defining region is also another fundamental step to perform segmentation. In a given color space, a region can be defined in four different ways:

- (Type one) A region can be defined by a class membership function in color space specifically for connected components of a pixel set. Their grouping is carried out in color space. A condition for grouping of pixels can be that the color of where they lie within a plane or on a solid surface in the color space.
- (Type two) A region is a connected set of pixels in the image plane for which it satisfies uniformity condition. The grouping of the color spectrum takes place in image plane instead of color space. It is usually created by merging other pixels or blocks of pixels or by splitting non-uniform regions.
- (Type three) A region is a connected set of pixels bounded by a color contour as a boundary. This color contour is determined by edge operator on a color image. The regions sometimes are uniform in uniformity measurement, since their boundary is created based on the non-uniformity measurement.
- (Type four) A region is a connected component of a set of pixels and their grouping is based on the physical model of the color signal in a color space. It is used to extract regions in the color image which corresponds to the surfaces of the object, and each

region consisting of one homogeneous material. Using these methods, shadow, shading or any other artificial color value changes on the image do not affect the segmentation result. This region definition is a basis for physics based image processing.

Depending on the region definitions, segmentation processes are divided into four classes. Pixel-based segmentation for the first type definition, area-based for the second type, edge-based for the third type and physics based for the fourth type [1, 3].

3.3.1 **Pixel-Based Segmentation**

A technique that employs type one region definition in color space is pixel-based segmentation. This segmentation technique doesn‟t consider spatial contexts rather it decides based on the content of the pixels. Its simplicity is its advantage. Pixel-based segmentation can employ two techniques: Histogram thresholding based technique and clustering technique.

3.3.1.1 **Histogram Thresholding**

Histogram thresholding is the simplest technique of pixel-based segmentation. The color histogram of an image with distinct regions will show clearly different peaks each of the peaks corresponding to different distinct regions. For instance, if an image has a distinct object on a background then the color histogram will have a deep valley. The bottom of this valley can be taken as a threshold so that pixels that belong to above and below this threshold value will be grouped in two different regions; this is called bi-level thresholding [1, 3]. Most of the time, medical images have many distinct regions and require multi-level thresholding. In this case, the histogram will have more than one deep valley and the selection of the thresholds will be about detecting valleys [1 - 6].

Color histogram is an interesting topic and has been used in various applications. As colors come in multiple channels/bands, their histogram computation is less straight forward compared to histograms of monochromatic images. One way of computing color histograms is to first combine the multiple channels (assume r, g, b) to a scalar valued two-dimensional function as proposed in [1] and given by:

$$\boldsymbol{F(r,g,b) = \frac{(0.25r + 4g - b)}{b}} \qquad 3\text{-}5$$

The above formula described in equation 3-5 has to be applied at each pixel position on the color image under consideration. After the conversion, we then perform the same histogram thresholding techniques that we use for gray scale images. However, incorporating the multi-dimensional channels without the need for conversion to a scalar should result in better segmentation results. After selecting suitable color spaces (for example RGB and HSI) the histogram of each color component in the chosen color spaces can be plotted. Then the most dominant peak of the six histograms determines the intervals of sub-regions. Pixels inside the interval create one region and pixels outside of the interval create another region. The selection of dominant peak from the histogram peaks of each component is determined by the priority of the components. For instance, the priority can be based on the number of determined frequencies (the higher frequency in the histogram will have higher priority) or a weighting value based on hue is high priority than brightness in segmentation [3, 4]. The color image segmentation can then be performed simply using histogram thresholding technique but selecting the effective color space is dependent on the application of the segmentation. Readers might want to consult previous works in [1, 3, 4, 5, 6] for further discussion on the method.

3.3.1.2 **Clustering**

The logical basis of the clustering technique is that the colors in an image tend to form clusters in the histogram, each cluster for each object in the image. The histogram is obtained by the color values of all pixels and the shape of each cluster is obtained. Then, each pixel in the image is assigned to the cluster that is closest to the pixel color. In this regard, there are different clustering algorithms which have been suggested in the literature including hierarchical clustering, fuzzy k-means clustering, etc. [5, 12]; however, because of the relevance and the scope of this document, only k-means clustering algorithm will be discussed in this document.

K-means algorithm: K-means clustering algorithm is one of the most popular and fastest clustering techniques. It is based on the minimization of performance index which is the sum of the squared distances from all points in a cluster domain to the cluster center where cluster centers define the midpoints of each of the classes to which the pixels should be assigned. The k-means clustering algorithm has the following procedures [1, 3]:

1. Determine K cluster centers $\boldsymbol{C_1(1), C_2(1), C_3(1), \ldots, C_K(1)}$ where $\boldsymbol{C_1(1)}$ is the color feature of the first cluster center during the first iteration and so on.
2. At the $\boldsymbol{k^{th}}$ iteration, each pixel $\boldsymbol{a}$ is assigned to one of the $\boldsymbol{K}$ clusters $\boldsymbol{C_1(k), C_2(k), C_3(k), \ldots, C_K(k)}$, where $\boldsymbol{C_j(k)}$ represents a set of pixels whose center is $\boldsymbol{C_j(1)}$. $\boldsymbol{a}$ is assigned to cluster $\boldsymbol{C_j(k)}$ if:

$$\|\boldsymbol{a - C_j(k)}\| \leq \|\boldsymbol{a - C_i(k)}\|, \qquad \text{3-6}$$

$$\forall \boldsymbol{i, j = 1, 2, \ldots, k}, \boldsymbol{i \neq j}$$

3. From the results of step 2, compute new cluster centers $\boldsymbol{C_j(k+1)}$, j = 1, 2, 3, . . ., **K** such that the sum of the squared distances from all points in $\boldsymbol{C_j(k)}$ to the new cluster center is minimized. In other words, the new

cluster center $\boldsymbol{C_j(k+1)}$ is computed so that the performance index $\boldsymbol{(J_i)}$ is minimized where:

$$\boldsymbol{J_i = \sum_{a \in C_j(k)} \| a - C_j(k+1) \|^2, \ j = 1, 2, 3, \ldots, K} \quad \text{3-7}$$

The new cluster center which minimizes this is the sample mean of $\boldsymbol{C_j(K)}$. Therefore, the new cluster center is given as:

$$\boldsymbol{C_j(k+1) = \frac{1}{N_j} \sum_{a \in C_j(k)} a, \ j = 1, 2, 3, \ldots, K} \quad \text{3-8}$$

where $\boldsymbol{N_j}$ is the number of pixels of cluster $\boldsymbol{C_j(k)}$.

4. If $\boldsymbol{C_j(k) = C_j(k+1)}$ for $\boldsymbol{j = 1, 2, 3, \ldots, K}$, the algorithm is converged and the segmentation procedure is terminated. Otherwise **Go to** step 2.

The crucial part of K-means clustering is the determination of cluster centers initially. If the initial centers are properly chosen, then the convergence of the algorithm will be fast. Therefore, cluster centers should be determined beforehand for the practical analysis of clustering. For example, cluster centers can be determined from the maxima in one-, two- or three- dimensional color histograms [1 - 6].

3.3.2 **Area-Based Segmentation**

Area-based segmentation uses criteria for homogeneity in the image area and it focuses on the continuity of a region. The objective is to produce connected regions that are as large as possible (i.e. to produce regions as few as possible). However, if the pixels are required to be too similar, then it will create over-segmentation. If the homogeneity value of pixels is too flexible, then more pixels will be taken as homogeneous and this will merge areas that had to be segmented. Therefore, such a trade of should be taken into account during area-based segmentation [1, 3, 5, 8].

The technique can be divided into region-growing and split-and-merge techniques. Region-growing technique is the process of grouping neighboring pixels based on their similarity and creates larger regions. It starts with "seed" pixels and then grows regions by adding neighboring pixels that have similar properties, for example, brightness, color, etc. to the seed. Then variance can be specified for the property chosen; the region growing stops when the pixel encountered is not in the specified variance range.

On the other hand, split-and-merge technique begins with non-uniform image areas. It splits up the area until uniformity criterion is reached. The split-and-merge technique splits the image into smaller and smaller regions and testing if the adjacent regions satisfy homogeneity criteria. If the adjacent smaller regions satisfy homogeneity criteria, then they will be merged [1, 3, 5]. Homogeneity can be defined in terms of the distances of pixel color values discussed in section 3.2.2.

A region is split into sub-regions until a given uniformity condition is achieved for the sub-regions. For gray-level images, this uniformity condition can be the variance of intensity levels within the sub-region that is smaller than a given threshold value. However, if the condition for the sub-region is not met, then this sub-region will be further divided in to smaller regions. Then finally, these adjoining images that are similar will be merged together according to a given condition of uniformity to get the uniform regions of maximum size possible [1, 3, 5]. This algorithm can be extended to vector-valued color images in which a color distance measurement is also included as a homogeneity condition. The use of color information enables the inclusion of perceptual attributes, like in using HSI color space for image segmentation [3].

3.3.3 **Edge-based Segmentation**

Edge-based segmentation techniques focus on the discontinuity of a region in the image, that means non-uniform or discontinuity measurements in the image are used for segmentation. The aim is to find a closed boundary such that the foreground is inside and the background is outside of this boundary. The technique can be classified as local or global. Local techniques use only the information in a pixel‟s local neighborhood for the detection of edge pixels. Global techniques use global optimization for the entire image and then identify edge pixels after several optimization steps. Often times such optimization techniques take long processing time [1, 3, 8].

3.3.4 **Physics-based Segmentation**

These are segmentation techniques in which physical models in a color space are used for the division of an image into regions that correspond to surfaces and/or objects in the scene. The goal of these techniques is to segment images of an object based on object borders and not being affected by the shades or reflections on the image. The mathematical methods in this technique are frequently similar to those discussed above [1, 3, 5, 8].

Since this thesis neither applies edge-based, area-based nor physics based segmentation techniques, the topics are not covered in detail. However, for further reading on these techniques, readers could consult previous works in [2 - 8].

3.4 Medical Image Segmentation

A meaningful image contains regions that convey information; therefore, identifying such regions (also called objects or regions of interest) from the background is the core purpose of extracting information from it. However, depending on the complexity of the image,

identification of objects in an image is not always easy and there are no one-size-fits-all techniques so far.

Different segmentation techniques have been developed, as some of them are described in previous sections. This thesis proposes a segmentation technique that uses both chromatic and luminosity difference measures as similarity measure for segmenting images. Moreover, the proposed segmentation technique analyzes each pixel rather holistically as opposed to serial monochromatic analysis commonly used by many other techniques. The next chapter discusses the proposed method in detail.

References

[1]Andreas K., Mongi A., “Digital Color Image Processing”, John Wiley & Sons, Inc., 2008.

[2]Mihai I., Noel R., and Dietrich P., ”Color Image Segmentation”, Springer Science, Business Media New York, Advanced Color Image Processing pp. 291-277, 2013.

[3]Konstantinos N. P., Anastasios N. V., ”Color Image Processing and Applications”, Springer-Verlag Berlin Heidelberg 2000.

[4]Rafael G., Richard W., “Digital Image Processing”, Prentice Hall, 3rdED., 2008.

[5] Henryk P., “Color Image Processing: Methods and Applications”, Taylor & Francis Group, LLC, pp. 103-128, 2007

[6]Hartigan, J.A., ”Clustering Algorithms”, John Wiley and Sons, USA, 1975

[7]Haralick R., Shapiro L., “Image Segmentation Techniques”, Computer Vision Graphics and Image Processing, 29,100–132, 1985.

[8]G. Dougherty, “Digital Image Processing for Medical Applications”, Cambridge University Press, 2009.

[9]Claus D. Tonnies, ”Guide to Medical Image Analysis: Methods and Algorithms”, Springer-Verlag London Limited, 2012

[10]Thomas M. Deserno, “Biomedical Image Processing”, Springer-Verlag Berlin Heidelberg, 2011

[11]Lim W., Lee U., ”On the color image segmentation algorithm based on the thresholding and the fuzzy c-Means techniques”, Pattern Recognition, pp. 1235-1252, 1990

[12] Charu C. A., Chandan K. R., “Data Clustering Algorithms and Applications”, Chapman & Hall/CRC, 2014

Chapter Four

4 Quaternion Based Medical Image Segmentation

4.1 Similarity Measurement of Color Pixels

Image segmentation refers to the process of partitioning an image in to several coherent and disjoint meaningful regions. The natural way to perform such image processing on colors is to represent pixels in a certain color space and compute relevant metrics [1, 2]. This processing could be done in a true color space (such as RGB) or in a transformed color space such as HSI. One important consideration is however that the color components represented in a chosen color space are essentially correlated with each other and this intrinsic information is often vital for further processing the colors [1 - 3]. It is believed that such a correlation could be highly non-linear; something that needs to be carefully exploited. This is so particularly when considering medical images. This means pixels of medical images have to be represented and analyzed holistically by keeping their color information intact as much as possible.

Intuitively, in order to segment images, their constituent pixels have to be clustered together depending on their likeness. Clustering pixels of an image is one of the pixel based segmentation techniques discussed in the previous chapter that relies on grouping pixels based on their similarity measure. In gray scale image segmentation, it is common to use intensity levels of pixels for similarity measurement. Even for color images, uniformity level of pixels are defined based on the gray intensity levels of their three color components [3]. Using such clustering techniques for color image processing is known to be ineffective, since it undermines the non-linear correlations among color components [2, 5]. Therefore, color pixels have to be first

represented properly and holistically so that computing their similarity will be more accurate. The accuracy of the similarity measurement used determines the effectiveness of the segmentation [1 - 4]. For this reason, this thesis proposes a different pixel representation and similarity measurement scheme. Once this measurement scheme is applied, its results will be used to cluster the pixels so that the image will be segmented properly using spectral clustering.

Since a color pixel contains more than one color component, each of which are real valued, its holistic representation can be accomplished vectorially using higher dimensional algebraic systems. One such a system that proved useful in color analysis makes use quaternions. By defining the similarity measurement in the quaternion space and then generating affinity matrix or affinity graph using this similarity measurement, a quaternion based spectral clustering method (QSCM) [2] can be derived for use in clustering of color pixels on medical images. The quaternion representation of RGB color pixels can be written as shown in equation 2-15. That means an RGB color pixel at position $(\boldsymbol{x, y})$ can be represented holistically as a single pure quaternion number $\boldsymbol{p(x, y)}$ as also done in previous works [6, 7]. This representation allows color pixels to be analyzed holistically as one entity without affecting the correlation of the color components of the pixels [2, 5, 6]. This technique will be more effective than monochromatic analysis of color pixels, especially to those images that have low intensity variations between pixels [7] making it particularly suitable for effectively analyzing medical images.

After representing a color pixel as a vector, the best way to rotate it in three-dimensional space is using quaternion rotation discussed in section 2.1.2. This rotation (for example, the rotation of $\boldsymbol{p}$ about an axis $\boldsymbol{q}$ through angle $\boldsymbol{2\theta}$) can be used in defining chromatic similarity measurements of color pixels [2] as shown in equation 2-10. If $\boldsymbol{q = e^{\mu\theta}}$, then $\boldsymbol{p}$ will be rotated by an angle

of 2θ. If we set q **a** unit pure quaternion, i.e. $\mu = \frac{(i+j+k)}{\sqrt{3}}$, this results in $\theta = \frac{\pi}{2}$. With these settings, the rotation of any quaternion about a pure unit quaternion q will be through an angle π or 180^0 [2, 8]. That means q is serving as an axis for the rotation with its three components equal i.e. $\frac{1}{\sqrt{3}}$**.** In RGB color space, a point on the gray line has equal component values (refer back to section 2.2.2). The rotation of a color pixel p in RGB color space about a gray line i.e. a pure unit quaternion q, creates a new quaternion $QR(p)$ point in the RGB space. This new point will be equal in magnitude but 180^0 opposite across the gray line. Therefore, the addition of p and $QR(p)$ will result in a quaternion number on the gray line. The summation of a pixel with another 180^0 rotated pixel can be used for measuring similarity between the two pixels [2]. Let‟s explain this in more detail. For example, to compute the similarity measure between two RGB pixels represented in the quaternion space say as $\mathbf{p_1 = r_1 i + g_1 j + b_1 k}$ and $\mathbf{p_2 = r_2 i + g_2 j + b_2 k}$ using quaternion rotation, first we can rotate either $\mathbf{p_1}$ or $\mathbf{p_2}$ around the pure unit quaternion **q**, through 180^0. Let‟ say we want to rotate $\mathbf{p_2}$ which can easily be done by computing $\mathbf{p_2 q p_2^*}$ where $\mathbf{p_2^*}$ is conjugate of $\mathbf{p_2}$**.** Then when $\mathbf{p_1}$ is added to $\mathbf{p_2 q p_2^*}$**,** the result will be another pure quaternion $\mathbf{p_3} = \mathbf{p_1} + \mathbf{p_2 q p_2^*}$ and any of the followings could hold:

- If $\mathbf{p_3}$ lies on the gray line, that means $\mathbf{p_1}$ and $\mathbf{p_2}$ have the same chromatic values i.e. **r**, **g** and **b** values of $\mathbf{p_1}$ and $\mathbf{p_2}$ are the same;
- If $\mathbf{p_3}$ is close to the gray line, that means $\mathbf{p_1}$ and $\mathbf{p_2}$ have close chromatic values;
- If $\mathbf{p_3}$ is far from the gray line, that means the chromatic values of $\mathbf{p_1}$ and $\mathbf{p_2}$ are different.

The addition of two pure quaternion numbers is again a pure quaternion number; therefore, $\mathbf{p_3}$ will also be: $\mathbf{p_3 = r_3 i + g_3 j + b_3 k}$ [2, 9].

$$\Delta \boldsymbol{QR}(\boldsymbol{p_1}, \boldsymbol{p_2}) = \left(r_3 - \frac{1}{3}(r_3 + g_3 + b_3)\right)\boldsymbol{i} + \left(g_3 - \frac{1}{3}(r_3 + g_3 + b_3)\right)\boldsymbol{j} + \left(b_3 - \frac{1}{3}(r_3 + g_3 + b_3)\right)\boldsymbol{k} \quad \text{4-1}$$

Equation **4-1** tries to normalize after computing the chromatic difference of pixels [1, 9, 18]; However, chromatic difference is not enough to measure the visual difference between two pixels. For example, the chromatic contents of bright and dark green pixels are identical but that doesn"t show how green they are. To make measurement of color pixel differences complete, the calculation has to include the luminance of the color pixels [2, 7, 9]. Therefore, the squared color distance or difference between $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$, $\boldsymbol{d}$ can be computed as the weighted sum of the square of the chromatic and luminance differences between the two pixels [9]:

$$\boldsymbol{d}(\boldsymbol{p_1}, \boldsymbol{p_2}) = \boldsymbol{t}|\Delta \boldsymbol{QR}(\boldsymbol{p_1}, \boldsymbol{p_2})|^2 + (\mathbf{1} - \boldsymbol{t})|\Delta \boldsymbol{I}(\boldsymbol{p_1}, \boldsymbol{p_2})|^2 \quad \textit{4-2}$$

where $\boldsymbol{d}(\boldsymbol{p_1}, \boldsymbol{p_2})$ is the color distance or difference between two pixels, $\boldsymbol{t}$ signifies the importance of chromatic and luminance variations between pixels and $\boldsymbol{t} \in [\mathbf{0}, \mathbf{1}]$. If the chromatic and luminance variations between the two pixels is equally important, then $\mathbf{t = 0.5}$. However, we can ignore $\mathbf{t}$"s effect if the two have equal importance. The two quaternion modulus terms in equation [4-4] can be calculated as discussed in chapter two. $\Delta \boldsymbol{I}(\boldsymbol{p_1}, \boldsymbol{p_2})$, which is the luminance variation between the two pixels $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$ can be defined as:

$$\Delta \boldsymbol{I}(\boldsymbol{p_1}, \boldsymbol{p_2}) = \mathbf{0.299}(\boldsymbol{r_2} - \boldsymbol{r_1}) + \mathbf{0.587}(\boldsymbol{g_2} - \boldsymbol{g_1}) + \mathbf{0.114}(\boldsymbol{b_2} - \boldsymbol{b_1}) \quad \textit{4-3}$$

where $\mathbf{0.299}, \mathbf{0.587}$, and $\mathbf{0.114}$ are coefficients that determine the contributions of red, green and blue respectively on luminance as shown in a previous study [9].

Using equations 2-10, 2-15, 4-1, 4-2 and 4-3, any two image pixels (both gray and color pixels) can be represented holistically and their similarity could be measured without the need to separate color components.

4.2 Medical Image Segmentation Using Quaternion Based Spectral Clustering

Images are created with pixels that can be represented in RGB color space and medical images are not different. Segmenting these pixels so that meaningful information can be extracted will require clustering of the pixels based on their similarity. There have been many data clustering algorithms, but non-are best suited to all. Among the data clustering algorithms, spectral clustering outperforms traditional clustering algorithms because of its relative simplicity for implementation [2, 4]. Therefore, in this thesis, spectral clustering is used to effectively and efficiently cluster pixels represented holistically in a chosen color space.

4.2.1 **Spectral Clustering**

Data sets can be difficult to cluster with traditional clustering techniques. For example, different synthetic datasets, shown in Figure 4-1, were processed by traditional clustering algorithms and they were proved to be difficult [11]. Such traditional clustering algorithms, that use Euclidean distance metric, for example, assume the clusters are „convex"; however, not all data clusters are convex. Evidently, such assumptions affect the result of the classical clustering algorithms whenever the datasets are not convex. Spectral clustering exhibits superior performance for both non convex and convex datasets [12].

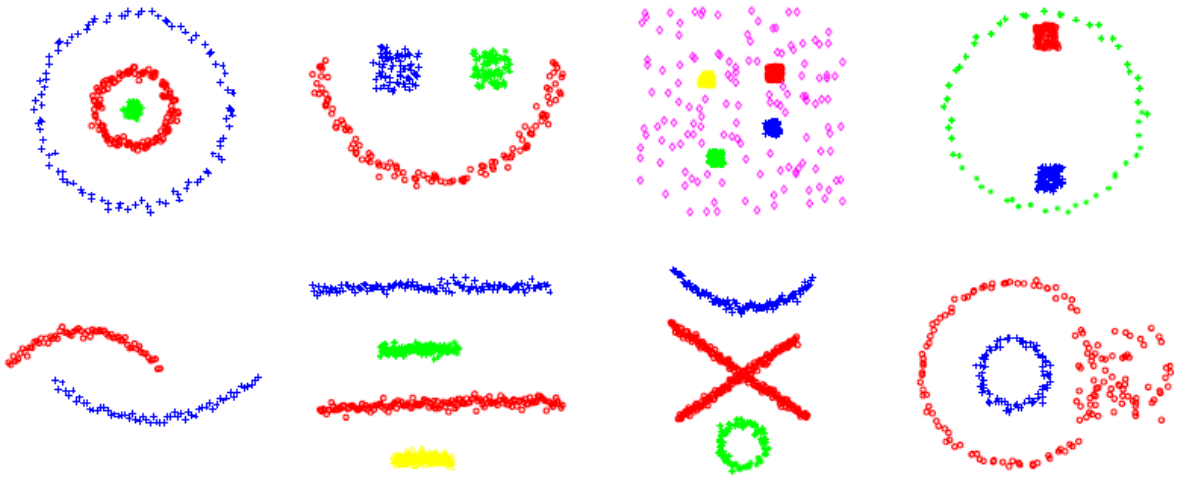


Figure 4-1 Self tuning spectral clustering on synthetic datasets

(Courtesy of Zelnik-Manor et. al, Advances in Neural Information Processing Systems, 17:1601–1608, 2004.)

Spectral clustering has been applied in image segmentation, text mining, speech processing and general data analysis [11 - 15]. The word spectral is used to denote that clustering results are obtained by analyzing the spectrum of the graph laplacian generated from the datasets.

Generally, a spectral clustering scheme can be described as a three step algorithm:

Step 1. Construct the similarity graph:

To partition a set of data points denoted by $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_n}\}$ into $\boldsymbol{k}$ subsets using spectral clustering algorithm, first the data set has to be represented as undirected "similarity graph", $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$. Each data point $\boldsymbol{x_i}$ is represented as a vertex $\boldsymbol{v_i}$ and E represents the edges between vertices. The nonnegative weighted $\boldsymbol{n}$-by-$\boldsymbol{n}$ *adjacency* (or *affinity*) matrix $\boldsymbol{W} = \{\boldsymbol{w_{ij}}\}_{\boldsymbol{i,j=1,\ldots n}}$ can be used to describe $\boldsymbol{G}$. There are different ways of generating affinity matrix like *k-nearest neighborhood graphs*, *ϵ-neighborhood graph*,

but for this thesis fully connected graph is preferred, which is discussed in detail in the next section.

Step 2. Embed the data points in a space with the use of eigen vectors of the laplacian graph:

After obtaining the similarity graph $\boldsymbol{G}$, computing the *graph laplacian matrices* is the next main step for spectral clustering. There is no unique way to compute graph laplacians but generally there are two types namely, *normalized* and *unormalized* graph laplacians. This thesis implements the *normalized* graph laplacian, which is discussed in detail in the next section.

Step 3. Apply a classical clustering algorithm, like k-means clustering, on the eigen vectors:

The final core stage of spectral clustering algorithm is applying classical clustering technique on the eigen vectors generated in step 2 above. In this thesis *k-means clustering* is selected for clustering the eigen vectors, which is again discussed in detail in the next section.

Readers may consult previous works in [11-15] for further information on spectral clustering.

4.2.2 The Proposed Image Segmentation Technique using Quaternion Based Spectral Clustering

Creating the similarity graph (or affinity matrix) is the core stage in spectral clustering which determines the efficiency of the clustering technique. As said before, the RGB color pixels are represented as a pure quaternion number in the quaternion space as shown in equation 2-15. After representing color pixels as pure quaternion numbers, the similarity measures of two pixels

can be computed based on chromatic and luminosity similarity measures using equation **Error! Reference source not found.** and equation 4-3. Then equation 4-2 is used to compute the holistic similarity of any pixels. The efficiency of clustering algorithm mainly depends on measuring this similarity. Finally, it is possible to adopt the spectral clustering algorithm to cluster color images based on their affinity matrix as described in this section.

Following the above premises, the proposed quaternion-based spectral clustering algorithm is explained in detail below while inputs from a previous work in [2] have been used during the design of the methodology.

1. Given a color image I (in RGB), we first need to map it to a quaternion valued image following the holistic pixel representation scheme discussed earlier using equation 2-15. If the image is gray, the three imaginary values of equation 2-15 will have the same value as the gray value.
2. The values used to measure the similarity between two pixels, $\boldsymbol{d(p_i, p_j)}$, shown in equation 4-2 are computed using quaternion rotation as shown in equation 2-10 and luminosity difference as shown in equation 4-3.
3. In this work, the Gaussian width $\boldsymbol{\sigma(i,j)}$ is assumed to be inversely dependent on the intensity variation between two pixels $\boldsymbol{p_i}$ and $\boldsymbol{p_j}$. The width $\boldsymbol{\sigma(i,j)}$ determines how fast the Gaussian function is rising and decaying. This made choosing a weighting value to be dependent on $\boldsymbol{\sigma(i,j)}$, in other words on intensity.
    3.1. As long as the two pixels differ in luminosity, that means they are different. For instance, let us take one pixel dark green and the other bright green. These two pixels have identical color (hue) content but they have different luminosity. Therefore,

the $\boldsymbol{\sigma(i,j)}$ has to be smaller and in effect the Gaussian function will generate higher weighting value. On the other hand, if one pixel is very dark green and the other pixel is very dark blue, their luminescence variation is small. In such very low luminosity difference situations, there is hardly a visual difference between the pixels and it is reasonable to consider them as the same and in effect a bigger $\boldsymbol{\sigma(i,j)}$ can be selected. In this case the Gaussian will generate smaller weighting value. Note again that in the proposed technique, the weighting function is dependent on the luminosity difference of pixels.

3.2. The range of values of $\boldsymbol{\sigma(i,j)}$ has to be bounded; otherwise, the weighting values might approach infinity. Meaningful values for $\boldsymbol{\sigma(i,j)}$ can be generated using equation 4-4 below.

$$\boldsymbol{\sigma(i,j)} = \frac{-\boldsymbol{\alpha}\Delta\boldsymbol{I}(\boldsymbol{p_i},\boldsymbol{p_j})}{(\boldsymbol{max}(\Delta\boldsymbol{I}) - \boldsymbol{min}(\Delta\boldsymbol{I}))} + \boldsymbol{\beta} \qquad 4\text{-}4$$

where: $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ are coefficients that need to be determined based on the type of the input image. $\boldsymbol{\sigma(i,j)} \neq \mathbf{0}$.

The values of $\boldsymbol{\sigma(i,j)}$ are proposed to be adaptive, depending on the luminosity difference between two pixels, $\Delta\boldsymbol{I}(\boldsymbol{p_i}, \boldsymbol{p_j})$.

3.3. Another reason for making the value $\boldsymbol{\sigma}$ dependent on the luminosity difference is that if the sum of the norm of the quaternion rotation (chromatic difference) and the norm of luminosity difference becomes equal for different pixels, the weighting matrix will be the same for both. However, if $\boldsymbol{\sigma(i,j)}$ computed using equation 4-4, then different pixels will be weighted with different values.

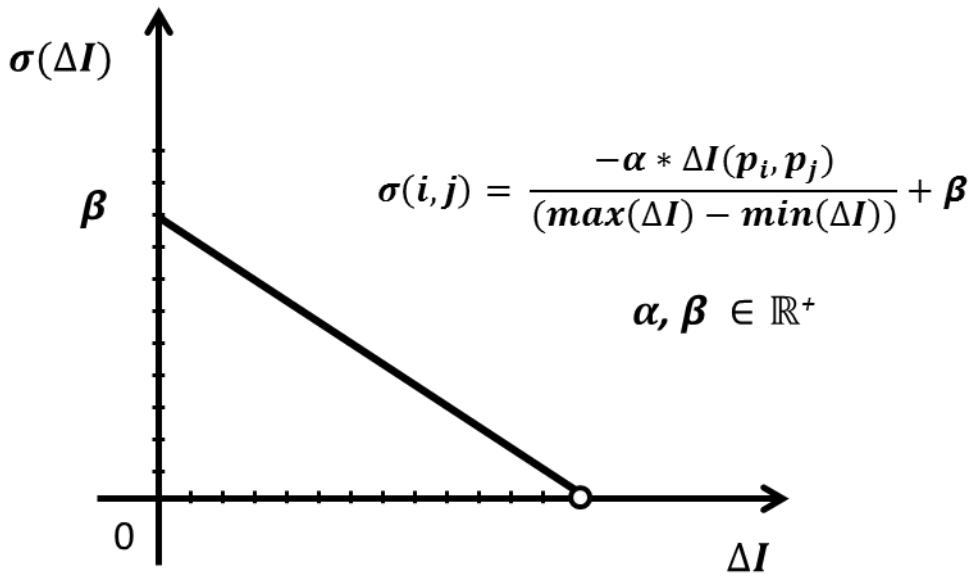


Figure 4-2 the plot of sigma function

The above Figure 4-2 shows that the width of the weighting values between pixels depends on the luminosity difference of the pixels. That is, if the luminosity difference of two pixels is large, then their weight has to be higher (the Gaussian will have sharp falling and rising sides) and vice versa.

4. Then build undirected and weighted graph **G** using the difference measure of two pixels. Then, the weighting function is chosen as Normalized Gaussian function as:

$$w_{ij} = \frac{1}{\sqrt{2\pi}\sigma(i,j)} e^{-\frac{d(p_i,p_j)^2}{2\sigma(i,j)^2}} \quad 4\text{-}5$$

where: $d(p_i, p_j)$ is the measure of difference between pixels p_i and p_j as discussed earlier and $\sigma(i,j)$ is the Gaussian. Equation 4-5 generates a symmetric affinity matrix W with non-negative entries where $w_{ij} = w_{ji}$.

5. Next we construct a diagonal matrix $\boldsymbol{D}$. This diagonal matrix is computed from the affinity matrix ($\boldsymbol{W}$) first by calculating the „degree matrix". The degree matrix is computed for each pixel as the sum of all similarity values connected to the pixel given by equation 4-6.

$$\boldsymbol{d_i} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{w_{ij}} \qquad 4\text{-}6$$

where: $\boldsymbol{d_i}$is the degree of a vertex (in our case a pixel) $\boldsymbol{i}$, $\boldsymbol{w_{ij}}$ the similarity or weighting value between pixel $\boldsymbol{i}$ with pixel $\boldsymbol{j}$. The diagonal matrix $\boldsymbol{D}$ is then constructed by taking $\boldsymbol{D_{ii}} = \boldsymbol{d_i}$ and zero otherwise.

6. Then using the diagonal matrix $\boldsymbol{D}$ and the affinity matrix $\boldsymbol{W}$, a normalized, symmetric laplacian matrix is constructed using the formula in equation 4-7.

$$\boldsymbol{L} = \boldsymbol{D}^{\frac{-1}{2}}\boldsymbol{W}\boldsymbol{D}^{\frac{-1}{2}} \qquad 4\text{-}7$$

7. The next step computes the eigen values $\boldsymbol{\lambda_i}$ and the corresponding eigen vectors $\boldsymbol{x_i}$ of the laplacian matrix $\boldsymbol{L}$ above. Note that since Equation 4-7 generates a symmetric laplacian matrix, then all eigen values are real valued.
8. We then sort the eigen values $\boldsymbol{\lambda_i}$ in decreasing order ($\boldsymbol{\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_n}$) and then compute the difference between consecutive eigen values, called gaps $\mathbf{g_i}$. List the gaps as row matrix, $\{\mathbf{g_1, g_2, \ldots, g_{n-1}} | \mathbf{g_i} = \boldsymbol{\lambda_i - \lambda_{i+1}}\}$.
9. Determine the number of clusters from the index of the first local maxima computed from the gaps.

$$\boldsymbol{if}\ \boldsymbol{g_1} > \boldsymbol{g_{j|j<i}}\ \&\ \boldsymbol{g_i} > \boldsymbol{g_{i+1}} \text{then}\ \boldsymbol{k} = \boldsymbol{i} \qquad 4\text{-}8$$

10. Stack the first $\boldsymbol{k}$ corresponding eigen vectors of $\boldsymbol{L}$ in columns to create a matrix $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_k}] \in \boldsymbol{R^{n \times k}}$.

11. Form matrix $\boldsymbol{Y}$ by normalizing each row of X:

$$\boldsymbol{y_{ij}} = \frac{\boldsymbol{x_{ij}}}{(\sum_j \boldsymbol{x_{ij}^2})^{\frac{1}{2}}} \qquad 4\text{-}9$$

12. Cluster each row of matrix $\boldsymbol{Y}$ using k-means clustering. Then assign the original pixel $\boldsymbol{p_i}$ into cluster $\boldsymbol{j}$ if and only if row $\boldsymbol{i}$ of matrix $\boldsymbol{Y}$ is assigned to cluster $\boldsymbol{j}$.

    Figure 4-3 presents the flow diagram of the proposed scheme summarizing all the above steps.

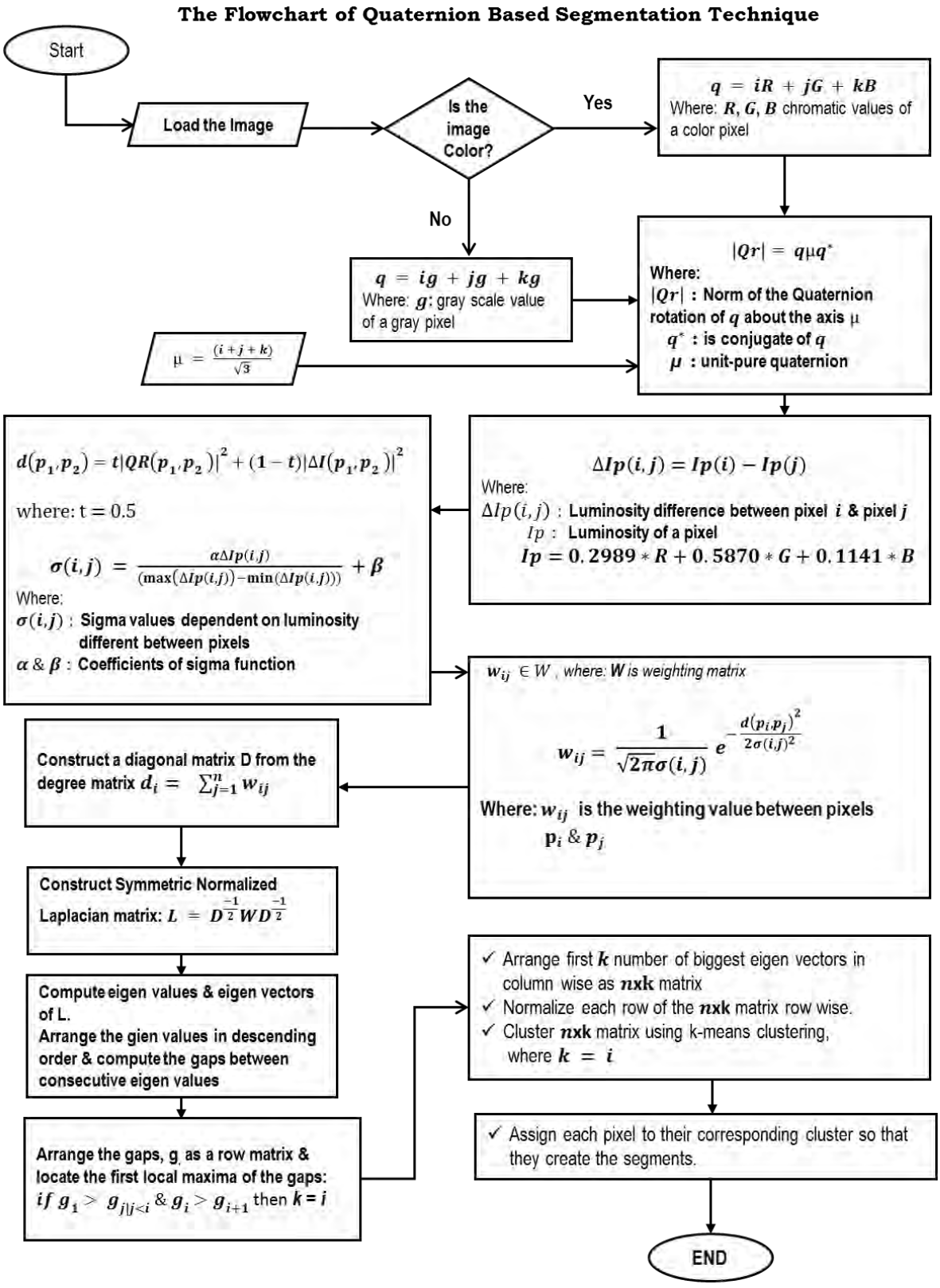


Figure 4-3 Flow chart of the proposed segmentation technique

4.3 Data Sets

Three data sets were used to test the the proposed segmentation technique. The first data set is composed of five synthetic color images generated using Matlab. The Matlab generated images are obvious enough to be segmented manually and are used as a proof of concept that the algorithm can perfectly do „the obvious". The second data set is composed of four publicly available medical images. These were composed of two color cancerous skin images taken by dermatologists as well as two retinal images generated from a fundus camera. Skin images with different degrees of difficulty including those subjects with low chromatic differences and intensity variations across the images were included within the samples taken [10]. These selections are believed to test the performance of the proposed technique well, since the technique mainly uses the chromatic and intensity values to determine the affinity matrix, ultimately the segmentation. The major difficulty with the fundus retinal images was that often objects of interest including exudates, the optical disc and others appear to have similar color with the background which undoubtedly makes clustering challenging [16]. The clustering results of the proposed algorithm applied on the retinal samples are included in the Appendix of this document. The performance of the proposed technique was also tested on two publicly available gray scale medical images. Particular interest was on T1 weighted contrast enhanced MR images of the brain taken from a cohort of patients diagnosed with the highest grade glioma cancer known by the name Glioblastoma multiforme [17].

References

[1] Cheng H., Jiang X., Sun Y., Wang J., "Color image segmentation: Advances and prospects", Pattern Recognition. 34(12), 2259-2281. 2001.

[2] Xiang L., Lianghai J., Hong L., Zeng H., “A quaternion-based spectral clustering method for color image segmentation”, SPIE 8003, 800303. 2011

[3] Zhu S., PlataniotisK., Venetsanopoulos A., “Comprehensive analysis is of edge detection of color image processing”, Society of Photo-optical Instrumentation Engineers. 38(4), 612-625. 1999.

[4] Ulrike V. L., “A Tutorial on Spectral Clustering”, Springer Stat. Comput. 17, 395–416. 2007.

[5] Dawit Assefa, et.al. “The Trinion Fourier Transform of Color Images”, ELSEVIER, Signal Processing 91, 1887–1900. 2011.

[6] J. Sangwine, "Fourier transforms of color images using quaternion or hypercomplex numbers", Electron. Lett. 32 (21), 1979-1980.1996.

[7]Ozlem N. S., Baba C. V., ”A Quaternion Framework for Color Image Smoothing and Segmentation”, Int J Comput Vis. 91, 233–250. 2011.

[8]William Rowan Hamilton, “Lectures on Quaternions”, Dublin, 1853.

[9]Lianghai J, Enmin S, Lei L, Xiang L., “A Quaternion Gradient Operator for Color Image Edge Detection”, IEEE, pp. 3040 – 3044, 2013.

[10] Publicly available color images of skin cancer types melanoma and squamous cell carcinoma was downloaded from http://www.cancer.org/cancer/skincancer/galleries/skin-cancer-images/ on May 12, 2016

[11] Charu C. A, Chandan K. R., "Data Clustering Algorithms", Taylor & Francis Group, LLC. pp.177 – 195, 2014.

[12] Lihi Z. M., Pietro P., "Self-tuning spectral clustering: Advances in Neural Information Processing Systems", 17:1601–1608, 2004.

[13] Jianbo S. Jitendra M., "Normalized Cuts and Image Segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8): 888 – 905, 1997.

[14] Inderjit S. D., "Co-clustering documents and words using bipartite spectral graph partitioning", Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 269–274, 2001.

[15] Francis R. B., Michael I. J., "Learning spectral clustering, with application to speech separation", Journal of Machine Learning Research, 7:1963–2001, 2006.

[16] Publicly available fundus images of patients with diabetic retinopathy was downloaded from https://www5.cs.fau.de/research/data/fundus-images/ on May 12, 2016

[17] Dawit Assefa, Harald K., David A. J., "Signal Analysis of Mutliparametric MRI images in higher order Fourier Spaces", International Journal of Computational Bioscience, Vol. 4, No. 1, 2013

[18] G. Elaiyaraja, N. Kumaratharan, C. Rama Prapau “Modified Quaternion Based Impulse Noise Removal with Adaptive Threshold from Color Video Sequences and Medical Images”, Middle-East Journal of Scientific Research 23 (7): 1382-1389, 2015

Chapter Five

5 Results and Discussions

The proposed quaternion based segmentation technique was developed using Matlab™ 2014b software on DELL OPTIPLEX 7010, CORE i7 with 16GB DDR3 RAM personal computer. The technique was tested on the data set acquired from different sources as explained in the previous chapter. Initially Matlab generated synthetic color images that are presumed easy for manual segmentation and noise free were considered. Next, the segmentation technique was tested on publicly available data sets comprised of color medical images of skins with lesions [1] and fundus generated retinal images [2]. Finally, the technique was tested on gray scale MR images of the brain taken from the data used a previous study [3].

5.1 Testing the Proposed Technique on Matlab Generated Color Images

Before testing the proposed segmentation technique on real medical images which are often times complex, it is logical to test the technique on predictable, noise free synthetic images. The primary purpose of using Matlab generated images for testing the technique was to quickly determine whether the result of the test is right or wrong based on inputs with obvious outputs. The first test is performed on a Matlab generated noise free color image which has three distinctly colored areas with colors Red, Blue and Green as shown in Figure 5-1*a)*.

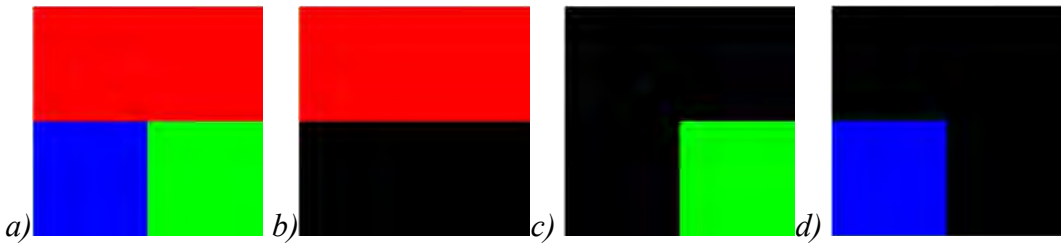


Figure 5-1 First test of the technique on Matlab generated color image and clustering results a) Matlab generated test image. b), c) and d) are the resulting segments of the proposed segmentation technique.

Figure 5-1 shows the test image with the results after applying the proposed segmentation technique. The optimum weighting function was obtained by selecting the coefficients of the sigma function, shown in equation 4-4, as $\boldsymbol{\alpha = 1.85}$ and $\boldsymbol{\beta = 2.85}$. Figure 5-1 b) c) and d) are the results of the technique which show the technique is able to generate the correct number of segments with perfectly segmented areas.

The technique was tested using another Matlab generated synthetic color image with four areas of the image colored with pure Red, pure Green, pure Blue and Magenta colors as shown in Figure 5-2 *a)*. In this case too, the same values of coefficients of the sigma function, shown in equation 4-4, i.e. $\boldsymbol{\alpha = 1.85}$ and $\boldsymbol{\beta = 2.85}$ were used. Figure 5-2*b), c), d)* and *e)* show that the technique was able to automatically generate the correct number of segments and the resulting four perfectly segmented areas.

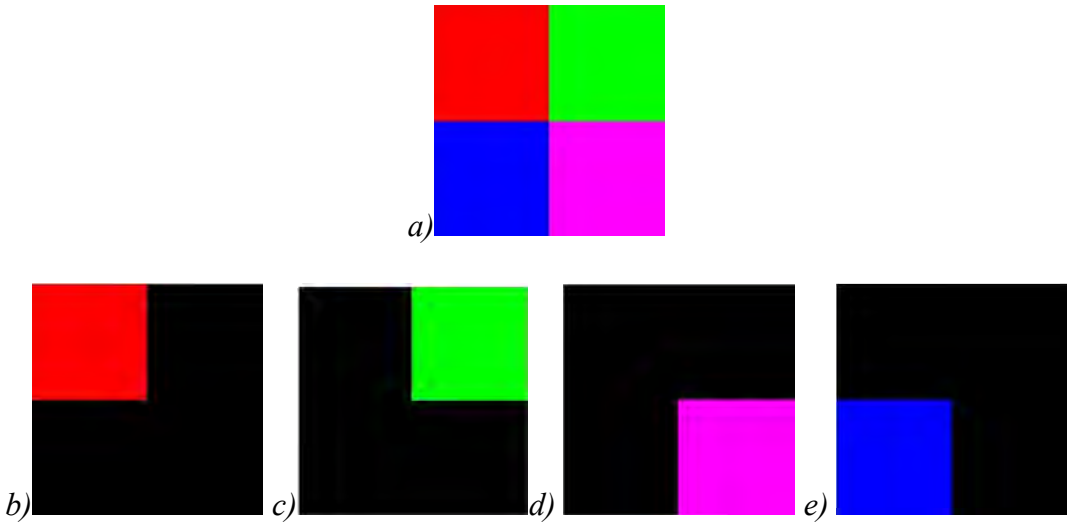


Figure 5-2Another test of the technique on Matlab generated color image and clustering results a) Matlab generated test image, b), c), d) and e) are the results of the proposed segmentation technique.

Then by increasing the number of colored areas of the test image, a third test image was created using Matlab. This time the image comes in six distinct regions and objects were in two different sizes: the left panel contains Green, Blue, and Cyan colors while the right panel contains Red, Yellow and Magenta colors as shown in Figure 5-3 *a).*

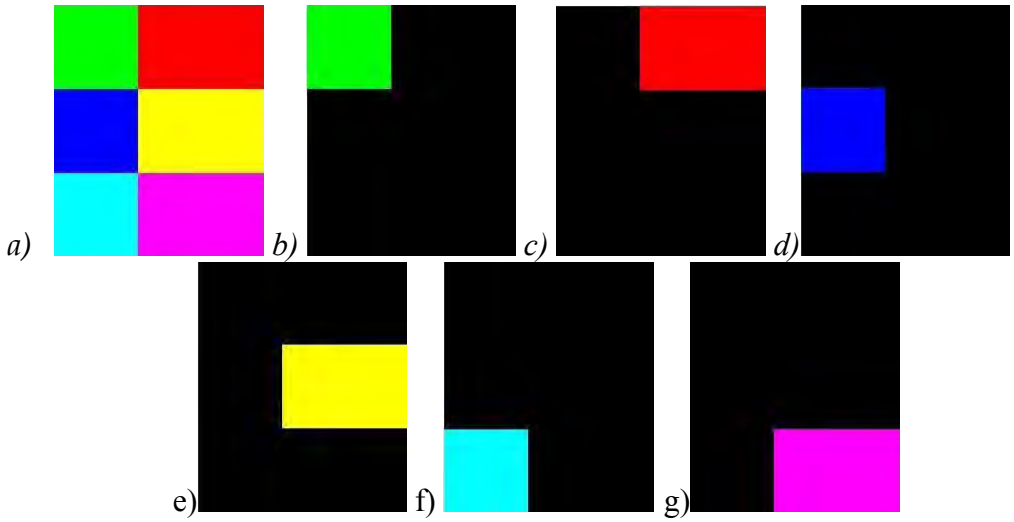


Figure 5-3 Third test of the technique on Matlab generated color image and clustering results a) Matlab generated test image. b), c), d) e), f) and g) are the results of the proposed segmentation technique.

The same coefficient values of the sigma function, shown in equation 4-4, i.e. $\boldsymbol{\alpha = 1.85}$ and $\boldsymbol{\beta = 2.85}$ were used as the previous cases. The clustered outputs shown in Figure 5-3*b), c), d), e), f)* and *g)* clearly show that the technique was able to automatically generate the correct number of segments and the resulting six perfectly segmented areas.

Another test image shown in Figure 5-4*a)* was synthetically generated on Matlab with eight distinct areas with Red, Green, Blue, Yellow, Cyan, Magenta, dark Red and dark Green subjects.

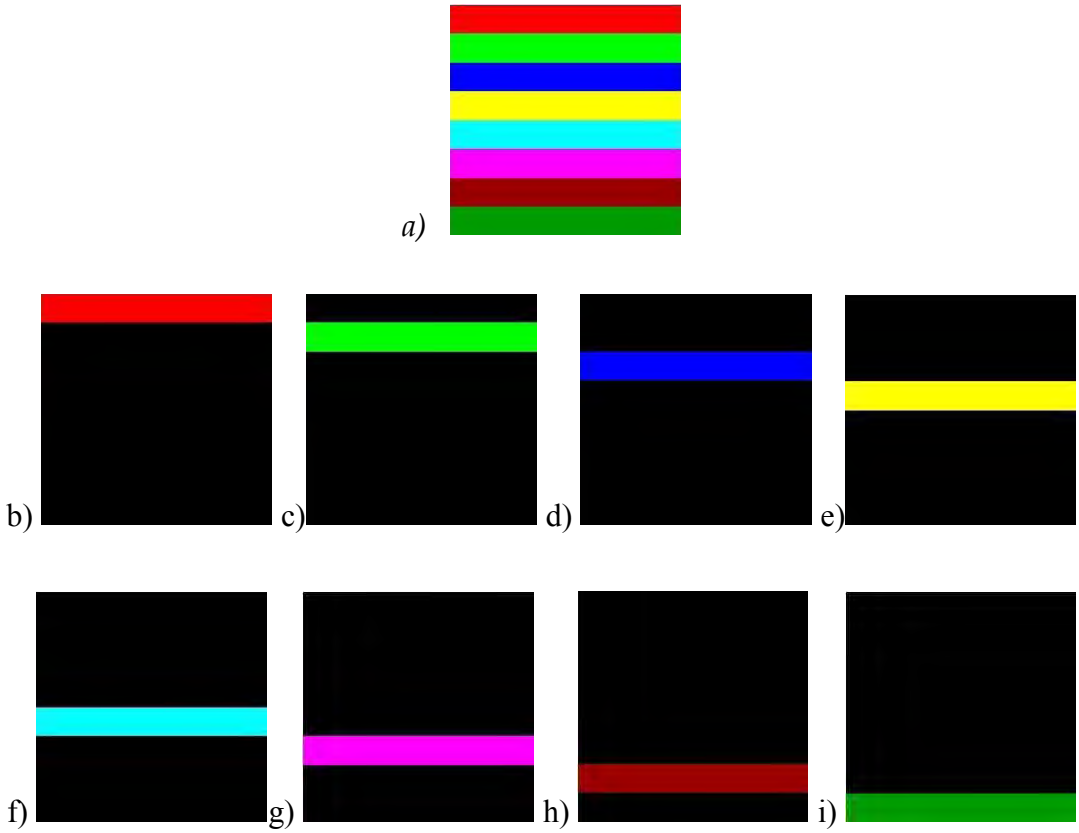


Figure 5-4 Fourth test of the technique on Matlab generated color image and clustering results a) Matlab generated test image comprising eight colors. b), c), d) e), f) g), h) and i) each color is segmented separately

Using the same coefficient values used previously, i.e. $\boldsymbol{\alpha = 1.85}and\boldsymbol{\beta = 2.85}$, the technique offered perfect clustering results as shown in Figure 5-4*b), c), d), e), f), g), h)* and *i).*

Another Matlab generated synthetic test image composed of four distinct color regions (Red, Green, Blue and Black) and with a different pattern is shown in Figure 5-5*a)*. The Blue and Green colored areas in the image are shaped as concentric circles while the Red colored area has a circular shape and the rest of the image is filled with pure Black pixels.

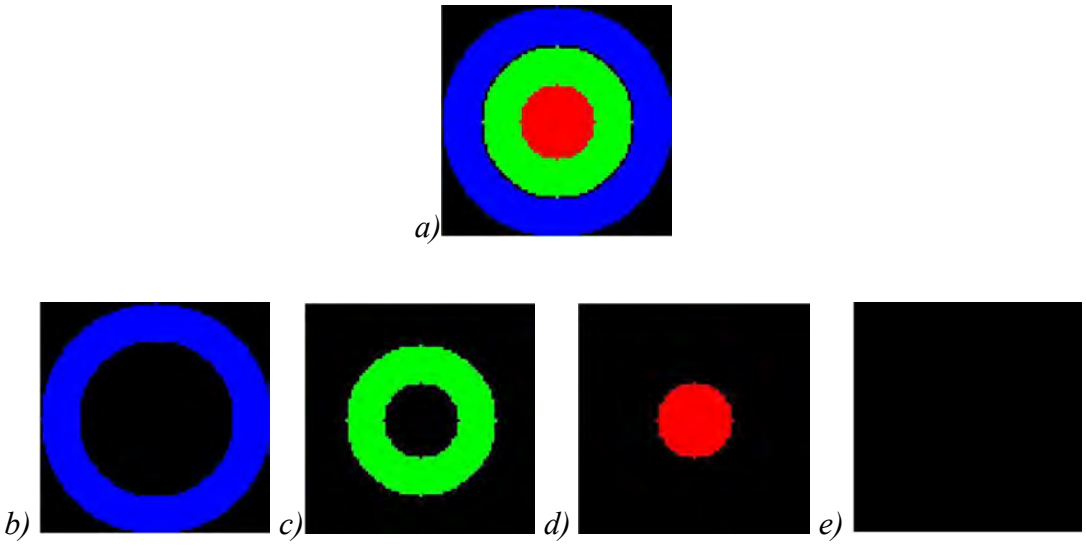


Figure 5-5Color image segmentation test 5 a) Matlab generated concentric circle test image. b), c), d) and e) are the results of the proposed segmentation technique.

Using the same coefficients for the sigma function ($\boldsymbol{\alpha = 1.85}$ and $\boldsymbol{\beta = 2.85}$**),** perfect clustering resulted as shown in Figure 5-5 *b), c), d)* and *e).*

The above test results presented here to demonstrate the effectiveness of the proposed quaternion based clustering scheme on synthetic color images generated on Matlab clearly demonstrated that the scheme can perfectly do the obvious with no ambiguity what so ever. It can automatically decide the number of clustered objects and segment them accordingly. When using holistic approaches like the proposed scheme in this thesis for use in color analysis, we need to make a clever use of the enormous amount of information embedded in the colors and the inter correlation between the color components. Clearly the developed scheme best utilized such vast information and this could have tremendous implications. We can think of other applications

such as automatic defect detection of colored subjects (eg. colored medical pills or other uniformly shaped colored products) with no bias. The scheme undoubtedly could add useful scientific values to the world of color image analysis in general.

One issue that worth mentioning here is regarding the sensitivity of the results with respect to the values of the coefficients of the sigma function (i.e. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) described in equation 4-4. The proposed quaternion based segmentation technique appeared fairly stable to changes in the values of the coefficients unless the coefficients in the weighting function deviate greatly from the described values ($\boldsymbol{\alpha = 1.85}$ and $\boldsymbol{\beta = 2.85}$).

5.1 Testing the Proposed Technique using Color Medical Images

After confirming the technique is able to segment the Matlab generated color images correctly, the next step was to test the technique on publicly available medical images [1]. However, due to the high computational demand of the technique during calculation of eigen-values and eigen-vectors, it was mandatory to reduce the sizes of the medical images under consideration by cropping out areas that are not required to be segmented and to work only over a region of interest.

The first medical image used for testing is a part of a skin lesion image that has minimum chromatic and luminosity variation, shown in Figure 5-7*a)*. It is obvious that the image comes with a great degree of difficulty and the testing of such images was believed to show the soundness of the proposed technique.

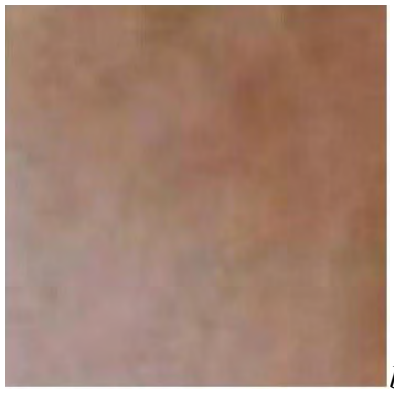
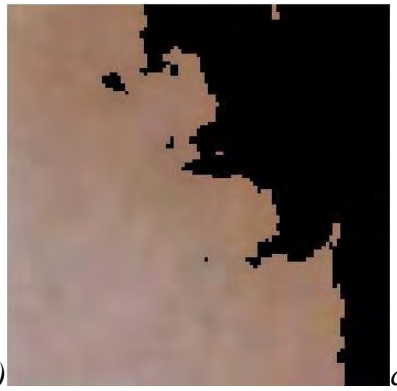
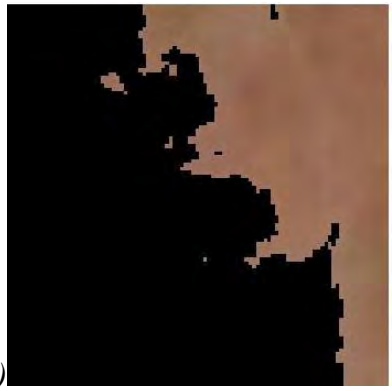
a)  b)  c) 

Figure 5-6 Testing the proposed segmentation technique on a part of skin image with lesion and the segmentation results

a) a part of skin image with lesion b) the segmented healthy part of the skin image c) the segmented lesion part of the skin image

The test image used, shown in Figure 5-6*a),* is difficult to segment manually because of its minimum variations in chromaticity and luminosity values; however, the technique was able to segment it in to healthy and lesion parts accurately as shown in Figure 5-6*b)* and *c)*. When it is observed closely, the test image has a less distinct boarder between the normal and lesion part of the skin. However, the technique was able to identify the border between the lesion and the normal skin in a way compatible with the visual judgment. The normal part of the skin is displayed in Figure 5-6*b)* and the lesion part of the skin is displayed in Figure 5-6*c).* The segmentation technique used for this test was optimized with values of coefficients of the sigma function described in equation4-4, as $\boldsymbol{\alpha = 0.85}$ and $\boldsymbol{\beta = 2.15}$. Coefficient values that vary greatly from these values make the output of the segmentation technique over-segmented or under-segmented.

The second medical image used to test the proposed segmentation technique is another skin image with distinct lesion shown in Figure 5-7*a).* The selected test image was from the area of the skin which has visually both distinct healthy and lesion parts. As it can be seen, the chromaticity and luminosity variation of Figure 5-7*a)* is different from that of previous skin

image shown in Figure 5-6*a)*; therefore, the weighting function is expected to be different. The optimum values of the coefficients of the sigma function are supposed to be different than the previous test skin image. However, the technique applied using the same coefficients as used in the previous skin image ($\boldsymbol{\alpha = 0.85}$ and $\boldsymbol{\beta = 2.15}$) offered promising results in identifying the lesion (cancerous) from the back ground as seen in Figure 5-7*b)* and *c)*.

c)

Figure 5-7 Testing the proposed segmentation technique on cancerous skin image

a) A cancerous skin image, b) the cancerous part of the skin after segmentation c) the healthy part of the skin after segmentation

Figure 5-7*b)* shows the lesion part of the skin and Figure 5-7*c)* is the healthy skin. Based on the chromaticity and luminosity values, we say the proposed technique segmented the skin image correctly because the normal skin and the two peaks of the swollen lesion have the same chromatic content and luminosity level. Once again the technique starts to fail if the coefficient values vary greatly from the mentioned values above.

5.2 Testing the Technique on Gray Medical Images

The holistic representation of pixels discussed in the previous chapter is designed not only to represent color pixels but also gray pixels. Equation **Error! Reference source not found.** represents color pixels as pure quaternion numbers; however, by making all the three components of the equation equal to the gray scale value, gray pixels can be represented accurately as a pure quaternion number. The rest of the computations needed to cluster gray scale images are exactly the same as those used to cluster colors. An example is presented in Figure 5-8*a*) as a demonstration. In this case a T1 weighted contrast enhanced MR image of part of the brain of a patient treated for glioblastoma multiforme (the enhanced area) is used.

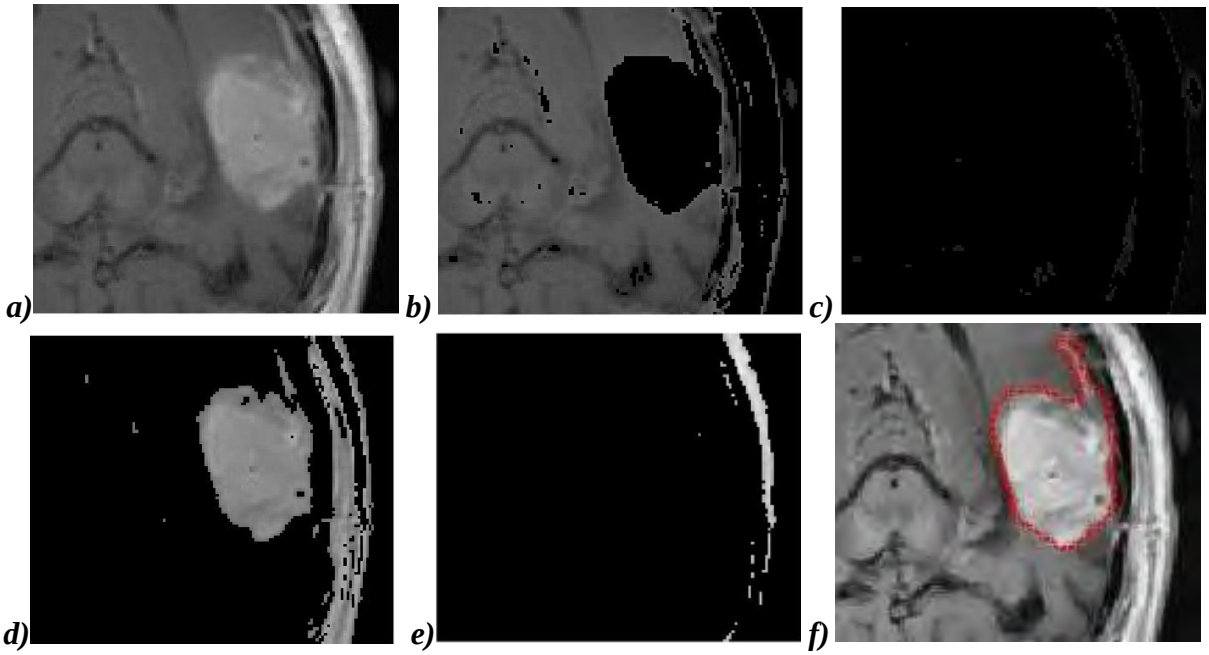


Figure 5-8 MRI image of a brain which is used for testing the proposed technique and the clustering results

a) MRI image of a brain with tumor b), c) d), and e) are the results of the segmentation technique. f) is the original T1 weighted image with the radiologist's contour drawn in red.

The optimum weighting function was obtained by selecting the coefficients of equation 4-4 as $\boldsymbol{\alpha = 2.0}$ and $\boldsymbol{\beta = 5.0}$. The results of the segmentation technique are shown in Figure 5-8*b), c), d), and e)*. It is interesting enough to see that the scheme correctly segmented the cancerous tumor and the back ground. Figure 5.8 *f)* shows the gold standard used following the delineation by an expert oncologist denoted on the image as a red contour. Figure 5-8*b)* shows the technique was able to isolate almost all of the gray matter of the brain and the edges of the skull successfully. Figure 5-8*d)* shows the technique was able to isolate the entire tumor. We clearly witness some difficulty by the method around the skull boundary. Well the intent in this example was to cluster different objects (focusing on the cancerous tumor) within the brain. The skull is not part of the brain. Of course, in most MR image processing applications, we first segment out the skull boundary and do further processing on the brain only and there are effective brain segmentation tools out there for this purpose. No effort was exerted to do that in this thesis as these subjects could simply be removed through simple visual assessment. However, it should be noted that the scheme mis-clustered the tumor with the skull boundary.

Another MR scan of a different patient treated for glioblastoma multiforme is presented in Figure 5-9*a)* showing a T1 weighted contrast enhanced tumor surrounded by a normal brain tissue and the skull boundary on the left of it.

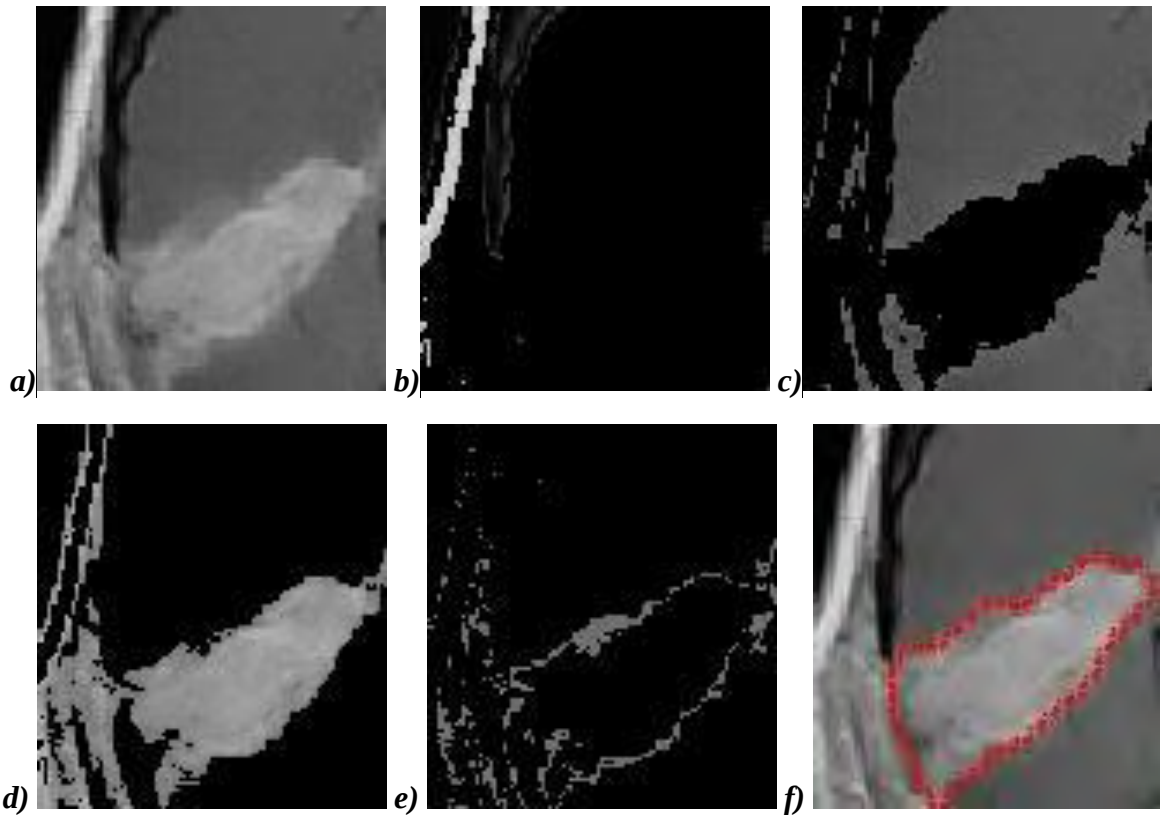


Figure 5-9 MRI image of a brain which is used for testing the proposed technique and the clustering results

MRI image of a brain with tumor b), c) d), and e) are the results of the segmentation technique. f) is the original T1 weighted image with the radiologist's contour drawn in red.

The optimum weighting function was obtained by selecting the coefficients of equation 4-4 as $\boldsymbol{\alpha = 1.80}$ and $\boldsymbol{\beta = 3.80}$. The results of the segmentation technique are shown in Figure 5-8*b), c), d), and e).* The segmentation scheme correctly segmented the cancerous tumor and the back ground. Figure 5-9*f)* shows the gold standard used following the delineation by an expert oncologist denoted on the image as a red contour. It can be seen from the results in Figure 5-9*b* and Figure 5-9 *c)* that the scheme correctly segmented the tumor and most of the gray matter of the brain. The Appendix A of this document presents few demonstrating results showing the application of the proposed clustering scheme on retinal images of patients treated for Diabetic Retinopathy generated on a fundus camera.

5.3 Performance Evaluation of the Proposed Technique

The performance of the proposed technique was measured for Matlab generated color images using Tanimoto coefficients, T (i.e.$0 \leq T \leq 1.0$). Tanimoto coefficients measure the similarity of two data sets or images. To measure the performance of segmentation technique using Tanimoto coefficients (with values between 0.0 and 1.0), two input images are required: one input is the ground truth and the other input is the segmentation result of the proposed technique; further information can be found on [4, 5].

$$\boldsymbol{T(A,B)} = \frac{\sum \boldsymbol{A} \cap \boldsymbol{B}}{\sum \boldsymbol{A} \cup \boldsymbol{B}} \qquad 5\text{-}1$$

In this work, equation 5-1 is used to compute the similarity of the two images, A and B. If the two images are the same, then T will result 1.0. If the two images are completely different, then T will result a number close to 0. Based on this coefficient, it is possible to compute the accuracy of the proposed clustering tool by comparing it against the available ground truth. Table 1 lists Tanimoto values computed after the application of the proposed scheme on the synthetic colors as well as the gray scale MR images.

Test images used	Quantity of Image Segments	T
Synthetic Color Images	10	1.0
Gray Scale Brain MR Images	2	0.9551 (max)

Table 1 Results of the performance evaluation of the proposed segmentation technique based on Tanimoto Coefficient

As it is shown in Table 1, the results of the Tinamoto coefficients computed for the synthetic color images segmentation resulted in 1.0. Therefore, the proposed technique automatically

segmented the synthetic noise free color images perfectly. This is the maximum perfection that can be obtained from any segmentation algorithm.

However, it was a daunting task to manually search for the optimum coefficient values of the sigma function. In this work, manual trial-and-error technique was used to determine the values of the coefficients of the sigma function and the results obtained from the proposed technique are far from perfect. The performance of the technique was measured using the two brain MR images described in section 4.3 by varying the values of the coefficients of the sigma function ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$). The maximum average performance of the technique on the two images is shown in Table 1.

Image 1	**α**	**1.0**	**0.5**	**2.0**	**2.0**	**1.85**
	$\boldsymbol{\beta}$	**6.0**	**5.0**	**5.0**	**3.0**	**2.85**
	T	**0.9128**	**0.9015**	**0.9128**	**0.9260**	**0.9260**
Image 2	**α**	**0.5**	**0.5**	**2.0**	**2.0**	**1.85**
	$\boldsymbol{\beta}$	**6.0**	**5.0**	**5.0**	**3.0**	**2.85**
	T	**0.9551**	**0.9404**	**0.9551**	**0.9501**	**0.9350**

Table 2 Tinamoto Coefficients for different segmentation results (for different coefficient values of the sigma function) of the two MR images using the proposed technique.

Table 2 quantitatively shows that the performance (measured as Tinamoto coefficient) of the proposed technique depends on the values of the coefficients of the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$; as their values vary the performance of the segmentation technique also varies. Therefore, with such performance dependency on the values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the technique has to be augmented with machine learning algorithms in determining the optimum values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. But within the scope of this work, it can be concluded that the performance result of the proposed segmentation technique is a promising algorithm as alternative tool for segmenting color, gray, medical and

non-medical images; however, for better performance, further work is recommended. Only after performing such further works that performance comparison with other algorithms would be meaningful. The suggested further works are included in the next chapter.

References

[1] Publicly available color images of skin cancer types melanoma and squamous cell carcinoma was downloaded from http://www.cancer.org/cancer/skincancer/galleries/skin-cancer-images/ on May 12, 2016

[2] Publicly available fundus images of patients with diabetic retinopathy was downloaded from https://www5.cs.fau.de/research/data/fundus-images/ on May 12, 2016

[3] Dawit Assefa, Harald K., David A. J., "Signal Analysis of Mutliparametric MRI images in higher order Fourier Spaces", International Journal of Computational Bioscience, Vol. 4, No. 1, 2013

[4] D. C. Anastasiu, G. Karypis, "Efficient Identification of Tanimoto Nearest Neighborhood", IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2016

[5] Jesna M, K. Raimond, "A Survey on MR Brain Image Segmentation Using SOM Based Strategies", International Journal of Computational Engineering Research, Vol, 03, Issue, 12

Chapter Six

6 Conclusion and Future Works

6.1 Conclusion

The objective of this research was to develop an alternative technique that can automatically segment both color and gray medical images acquired using certain modalities. Undoubtedly this is one area of research that is interesting and useful with so many potential implications in the image processing world. The basis for the developed clustering tool in this study is holistic representation of color pixels which enables clever exploitation of the inter correlation information among the color bands/channels. This is as opposed to a serial analysis that requires separation of color components. It is the combination of the color channels that often give rise to color textures, patterns and other useful attributes of a color scene, and hence holistic representation of such color pixels could be quit vital. This thesis presented itself as a proof of concept that this hypothesis is indeed valid. The quaternion approach has been used for use in holistic representation of color pixels vectorially and the respective algebra (including quaternion rotation) has been used to develop a quaternion based clustering scheme. The various tests carried out to evaluate the performance of the proposed scheme clearly indicated that the method carries great potentials to be translated to the clinics supporting physicians and related health care workers from time consuming and costly visual assessment of huge volume of image data which is prone to subjectivity.

The proposed technique can be used as a perfect tool for segmenting noise free color images, as is shown in Table 1. However, medical images are never noise free and this could pose serious

challenges during image segmentation. Despite the performance of the proposed scheme when applied on selected color medical images; qualitatively the technique was shown to be promising. As quaternions allow representation of gray scale images as special types of colors, the proposed method has also been tested on monochromatic images particularly MR brain scans of patients with glioma tumors. Results generated were found to be very promising.

In many other clustering algorithms which are proposed in the literature, image pre-processing precedes a complete processing algorithm development. As we always say this could be a risky procedure particularly when the subjects are medical images. The tread-off between need for image pre-processing and data lose is quite critical when considering medical images. Such procedures are intentionally omitted in the present study. Nevertheless, carful pre-processing of the images could help in enhancing some objects of interest degraded due to noise or other artifacts. Particularly those medical images with predictable noise/artifact content could be allowed a pre-processing stage making them suitable for a more effective clustering work. Addressing such an issue needs further investigation.

6.2 Future Works

There are yet two major outstanding issues awaiting further research. One has to do with the selection of the two optimal parameters used to compute the sigma function. The other being the computational time required for the algorithm to cluster a given subject. As it stands now, the parameters used to compute the sigma function are picked manually. This could sound downplaying a potentially very useful algorithm. This procedure of parameter selection through trial-and-error is tedious as well as time consuming. Automating the process needs further research. This could pave a way for further study to be carried out on how to generate optimum

values of the coefficients of the sigma function by using machine learning approaches. Machine learning algorithms can be devised to generate values of coefficients of sigma function which are more likely to be specific to specific imaging modalities.

The computational aspect of the proposed algorithm is also worth further investigation. For instance, the size of the affinity matrix that is required to cluster a size of n data points (in our case the total number of pixels in an image) is $n \times n$. That means a size of 1024 $\times$ 1024 pixel color or grayscale image requires an affinity matrix of size (1024*1024)×(1024*1024) and the resulting matrix requires a computer with bigger memory and super-fast processing speed. The most time consuming session of the process is computation of the eigen values and corresponding eigen vectors required during the clustering procedure. In this study a function that is already built-in Matlab has been used for the purpose of computing the eigen values and eigen vectors. Optimizing this step could significantly reduce the processing time. Furthermore, though the weighting matrices play a crucial role during the clustering process, their dimension could significantly be lowered and yet be good approximations thereby increasing the computational speed of the process. As mentioned earlier, another topic that needs further study is the issue of incorporating a pre-processing step that doesn‟t necessarily result in considerable information loss. Moreover, only qualitative analysis has been presented in this thesis to evaluate the performance of the proposed scheme against the available ground truth. A quantitative evaluation of the method is essential involving different accuracy measures (sensitivity, specificity, overall accuracy and the like). Resolving these and similar other issues awaits further investigation.

References

[1] Xiang L., Lianghai J., Hong L., Zeng H., “A quaternion-based spectral clustering method for color image segmentation”, SPIE 8003, 800303. 2011

[2] Ulrike V. L., “A Tutorial on Spectral Clustering”, Springer Stat. Comput. 17, 395–416. 2007.

[3] Charu C. A, Chandan K. R., “Data Clustering Algorithms”, Taylor & Francis Group, LLC. pp.177 – 195, 2014.

7 Appendix A

Generating the values of the coefficients ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) of the sigma function through trial-and-error was more difficult while analyzing the retinal color images generated on a fundus camera. In this research it was proved that fundus images appeared difficult to be segmented using the proposed clustering algorithm. One likely reason for such difficulty is that the sigma function is defined to be linearly related with luminosity differences of pixels. A fundus image often comes with pronounced illumination variabilities across the image domain. This undoubtedly makes choice of the sigma function coefficients very difficult.

Three representative fundus retinal images are presented below for demonstration. All images were generated on a fundus camera from patients treated for diabetic retinopathy. The first example presents an image taken from a patient following laser treatment while the other two were taken before treatment. In the first image, as seen on Figure 7-1, the algorithm detected two dominant subjects: laser treated lesions and the background. The tiny blood vessels are not detected by the algorithm. The images presented in Figure 7-2 and Figure 7-3 both depict part of the fundus image around the Optic Disk (OD) with visible blood vessels emerging from it. Though with great difficulty, the algorithm detected the OD, the blood vessels as well as the rest of the background. Note that in diabetic retinopathy studies, accurate OD localization is considered very useful.

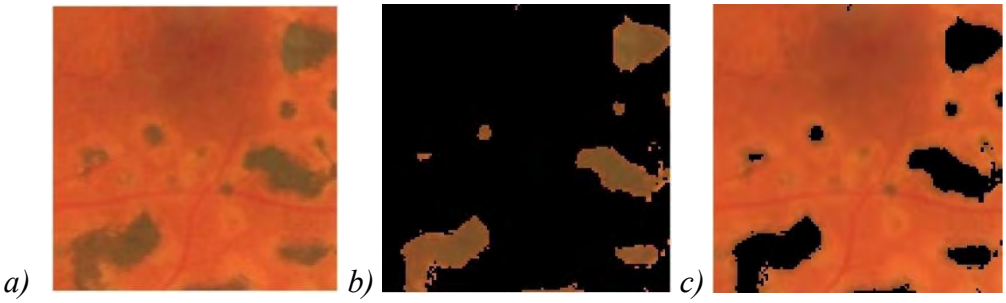


Figure 7-1 Testing the proposed technique using fundus image of a diabetic retinopathy patient with pigments developed after a laser treatment using values $\boldsymbol{\alpha = 1.15}$ *and* $\boldsymbol{\beta = 3.15}$

a) Fundus image with pigments developed following a laser treatment b) the segmentation result showing the pigments c) the segmentation result showing the rest of the retina

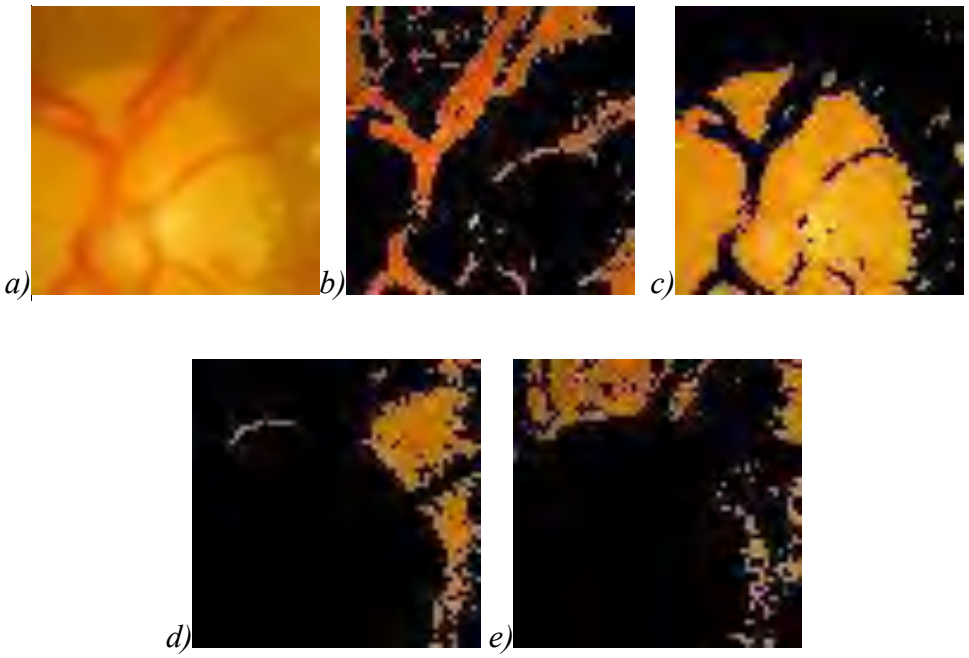


Figure 7-2 Testing the proposed technique using fundus image of a diabetic retinopathy patient cropped around the optic disc (OD) using values $\boldsymbol{\alpha = 8.0}$ *and* $\boldsymbol{\beta = 2.0}$
a) The result of segmentation technique showing the blood vessels b) the result of the segmentation technique showing optic disc of the retina d) and e) are the segmentation results showing the rest of the retina

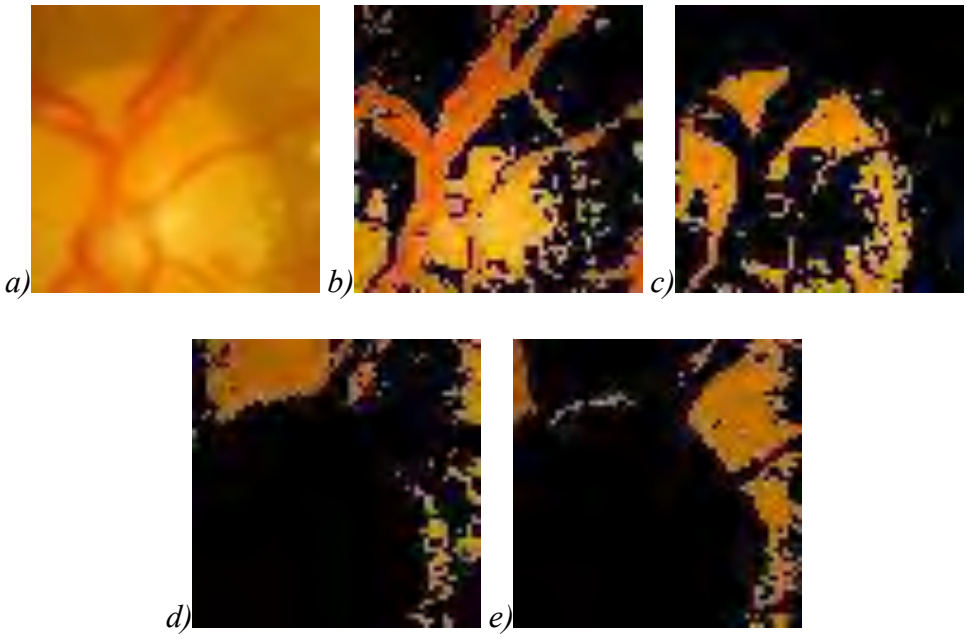


Figure 7-3 Testing the proposed technique using the fundus image presented in Figure 7-2 above but now using $\boldsymbol{\alpha = 8.0}$ *and* $\boldsymbol{\beta = 2.1}$

a) The result of segmentation technique showing the blood vessels with some part of the optic disc b) the result of the segmentation technique showing part of the optic disc of the retina d) and e) are the results showing the rest of the retina

As it is shown in Figure 7-2 and Figure 7-3, a slight change in the $\boldsymbol{\beta}$ coefficient (i.e. $\boldsymbol{\beta}$ = 2.0 and $\boldsymbol{\beta}$ = 2.1) has resulted in different clustered outputs. This demonstrates a case where the clustering outputs using the proposed algorithm are very sensitive to the two parameters that should be picked manually through trial-and-error.

8 Appendix B

The main Matlab Program used to implement the proposed segmentation technique:

```
% ======================================================

% Medical Image Segmentation Using Hypercomplex Analysis
% Samuel Getaneh Bayih, July 2014
% email: smlgtnh@gmail.com
% ======================================================
close all; clc; clear all;
%1. Loading the image
    imQ = imread('c:\MTLB\triedEx\Fundus.jpg');

    [m,n] = size(im(:,:,1));

%2. Reshaping the image data to make it column matrix by three
    im_Quat_col = zeros(m*n, 3, 'double');
    ncol = 0;
    for i = 1:n
          im_Quat = squeeze(imQ(:,i,:));
          im_Quat_col(ncol+1:ncol+m,:) = im_Quat;
          ncol = ncol + m;
    end;

%3. PURE-UNIT QUATERNION as axis of rotation...
    u = 1/sqrt(3)*[1,1,1];

%4. QUATERNION ROTATION about u
    QuaternRot = quaternRot(u,im_Quat_col);  % Quaternion rotation function
                                             % call

%5. Luminance of pixels
    Ip = 0.2989*im_Quat_col(:,1) + 0.5870*im_Quat_col(:,2) +
         0.1141*im_Quat_col(:,3);

%6. Difference of pixels
    [Dpixels,DIp] = DPix(QuaternRot, Ip); % Similarity measurement function
                                          % call

%7. Sigma Computation
    [sig] = sigma(DIp);                   % sigma function call

%8. Affinity matrix calculation

    [W_mat] = W(Dpixels, sig); % Normalized Gaussian Weighting function call
```

```
%9. Construct the Laplacian matrix...

    [laplacian_matrix] = Laplacian_mat(W_mat);  % Laplacian matrix function
                                                % call


%10. Eigen calculation
tic
    [eVec,eVal] = eig(laplacian_matrix); % eigen value and eigen vector
                                         % calculation function call

%11. Eigen value GAP calculation
    [eVec,eVal] = sortem(eVec,eVal); % sorting the eigen values & vectors
    [eVal_gaps] = Gaps(eVal);        % computing the eigen gaps

%12. Local max of gaps
     [pcks,locs] = findpeaks(eVal_gaps); % finding indices of local maximas

    i = locs(1);                     % The first local max
    XeVec = eVec(:,1:i);             % The first i(1) largest eigen vectors

%13. Row-wise Normalization of the first i(1) eigen vectors
     [p,q] = size(XeVec);
     YeVec = zeros(p,q);

    for id = 1:p
        YeVec(id,:) = XeVec(id,:)./sqrt(sum(XeVec(id,:).^2));
    end;

%14. Kmeans clustering
    idx = kmeans(YeVec,i);
%
    [rsz,csz] = size(im(:,:,1));
    fin_im = zeros(rsz,csz);

%15. Reconstructing using the pixels
    rw_cl = 1;
    for Kolem = 1:csz
      for Raw = 1:rsz
             fin_im(Raw,Kolem) = idx(rw_cl);
             rw_cl = rw_cl + 1;
      end;
    end;

% assigning the pixels to their cluster according to the k-means result
    clstr = max(max(fin_im));
```

```
    img_segmnts = zeros(m,n,clstr);
    for clNum = 1:clstr
          img_segmnts(:,:,clNum) = Imgs(fin_im,clNum);
    end;
% reconstructing the resulting clusters as segments
     img_segmnt(:,:,:) = uint8(img_segmnts(:,:,:));
     imshow(im);

    for ij = 1:clstr
        im_final(:,:,1) = im(:,:,1).*img_segmnt(:,:,ij);
        im_final(:,:,2) = im(:,:,2).*img_segmnt(:,:,ij);
        im_final(:,:,3) = im(:,:,3).*img_segmnt(:,:,ij);
        figure, imshow(im_final);
    end;
```