Color Image Edge Detection and Segmentation: A Comparison of the Vector Angle and the Euclidean Distance Color Similarity Measures

by

Slawomir Bogumil Wesolkowski

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 1999

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Slawomir B. Wesolkowski

Acknowledgements

I would like to thank Dr. Ed Jernigan for his continued support and help in the long years that lead to the preparation of this thesis. I am also grateful to him for allowing me to choose my own topic of study and for letting me carry out the research at my own pace.

I would also like to thank Dr. Bob Dony for the rich discussions on color image processing and in particular for the insight of using the vector angle as a color similarity measure. I would like to thank him, as well as Dr. Catherine Burns for taking the time to review this thesis and for providing very helpful comments.

To NCR belongs the credit of allowing me to attend the University of Waterloo on a part-time basis while I worked there full time and for the tuition reimbursement program which provided much needed financial support.

I would also like to acknowledge the warm support of my dear companion, my family and my friends in encouraging me to finish this work.

Last but not least, I would also like to thank Dr. Paul Fieguth for spending time in reviewing part of this work, as well as Dr. Stewart Kelland and Dr. Lowell Winger for their comments on early versions of this thesis.

Abstract

This work is based on Shafer's Dichromatic Reflection Model as applied to color image formation. The color spaces RGB, XYZ, CIELAB, CIELUV, rgb, $l_1l_2l_3$, and the new $h_1h_2h_3$ color space are discussed from this perspective. Two color similarity measures are studied: the Euclidean distance and the vector angle.

The work in this thesis is motivated from a practical point of view by several shortcomings of current methods. The first problem is the inability of all known methods to properly segment objects from the background without interference from object shadows and highlights. The second shortcoming is the non-examination of the vector angle as a distance measure that is capable of directly evaluating hue similarity without considering intensity especially in RGB. Finally, there is inadequate research on the combination of hue- and intensity-based similarity measures to improve color similarity calculations given the advantages of each color distance measure.

These distance measures were used for two image understanding tasks: edge detection, and one strategy for color image segmentation, namely color clustering. Edge detection algorithms using Euclidean distance and vector angle similarity measures as well as their combinations were examined. The list of algorithms is comprised of the modified Roberts operator, the Sobel operator, the Canny operator, the vector gradient operator, and the 3x3 difference vector operator. Pratt's Figure of Merit is used for a quantitative comparison of edge detection results. Color clustering was examined using the k-means (based on the Euclidean distance) and Mixture of Principal Components (based on the vector angle) algorithms. A new quantitative image segmentation evaluation procedure is introduced to assess the performance of both algorithms.

Quantitative and qualitative results on many color images (artificial, staged scenes and natural scene images) indicate good edge detection performance using a vector version of the Sobel operator on the $h_1h_2h_3$ color space. The results using combined hue- and intensity-based difference measures show a slight improvement qualitatively and over using each measure independently in RGB. Quantitative and qualitative results for image segmentation on the same set of images suggest that the best image segmentation results are obtained using the Mixture of Principal Components algorithm on the RGB, XYZ and rgb color spaces. Finally, poor color clustering results in the $h_1h_2h_3$ color space suggest that some assumptions in deriving a simplified version of the Dichromatic Reflectance Model might have been violated.

Table of Contents

Table of Tables

Table of Figures

Chapter 1: Introduction

1.1 Background

When humans are asked to describe a picture, they generally give a list of objects within the picture as well as their relative positions [20]. However, upon closer examination the image reveals object shadows, highlights from shiny object parts and differences in the color brightness of an object (i.e., object parts away from the light source appear darker than those closer to it). Shiny object parts could also produce object reflections in another object. Various physical processes by which light interacts with matter can explain these optical effects. An image understanding system that would generate descriptions similar in quality to the ones given by humans will have to discount the influence of these physical processes.

The purpose of a general image understanding system is to recognize objects in a complex scene or document image. Typically, one of the first steps in such a system is edge detection [20]. Edge detection algorithms usually detect sharp transitions of intensity and/or color within an image. These transitions are characteristic of object edges. Once edges of an object are detected other processing such as region segmentation, text finding, and object recognition can take place. Researchers have concentrated in the past few decades on devising algorithms for grayscale image understanding [28]. With the advent of powerful personal computers, it is now possible and practical to move to the more computationally intensive realm of color image understanding. There are many benefits in using color images. For example, the increase in the quantity of information can be used for more accurate object location, processing, and the possibility of processing images that are more complex.

The trichromatic model of color representation is fundamental to the human perception of color. A typical color image capturing system relies on a trichromatic input based on the additive primary colors red, green and blue [19,44]. This is commonly known as the RGB color space. The RGB model corresponds most closely to the physical sensors for colored light (e.g., the cones in the human eye), and it is implemented as red, green, and blue filters in most color CCD sensors. However, the human perception of color qualities can be said to follow more closely the HSI (Hue, Saturation, and Intensity) model [35]. This model creates a color image representation based on the amount of light (i.e., the Intensity), the amount of color (i.e., the Saturation) and color as described by the wavelength (i.e., the Hue). There exist a number of other equivalent color spaces with important properties. They will be discussed later in this thesis.

The perception of color is fundamental to the human visual system. It would make sense, therefore, to try to use this information to improve the accuracy of current grayscale algorithms. In color image understanding, similar tasks are performed as in the grayscale image world. The fundamental difference is the availability of chromaticity information such as hue and saturation for example. Several researchers have already used color images for complex applications: text finding [72], enhancing binary check documents [65], automatic granite inspection [50], automated road navigation [2], and color image map segmentation [24].

It would be useful to design a system that follows more closely how the human visual system perceives color objects. To that end, the physical processes that underlie the optical effects have been studied. In this thesis, the work presented is derived from a physics-based approach based on the Dichromatic Reflection Model introduced by Shafer in 1985 [51]. Physical processes involved in the image formation stage have not been a strong point of interest in the traditional line of computer vision research [29]. The common practice has been and still is to divide the image understanding process into two phases: a feature extraction phase and a higher-level object model-fitting phase. The former is characterized by a low-level segmentation process such as edge detection. In the latter, image features are related to object features described in object models of the scene. From this point of view, image processing has been considered a statistical image processing problem where the determination of statistically significant differences between pixel values under the presence of noise is a major concern. It is assumed implicitly that significant changes correspond to object boundaries in the scene.

However, the edges and regions thus generated will probably not only outline material boundaries (which is what image segmentation purports to accomplish), but also shadows, highlights, and intensity changes across the object. The segmentation result is then passed to a higher-level process that tries to merge regions across highlights and shadows according to the model representation of each object for the specific application.

1.2 Motivation

As computing power increases, more complex image processing operations can be implemented in real time applications. Color image processing has been researched for several decades, however, this research was more motivated by theoretical advances (such as devising more intuitive color spaces) than by practical applications [28,29,51]. Today, image processing applications based on binary or gray images are being introduced regularly. There is a great need to extend the image processing knowledge base to color-specific implementations of these algorithms and to new color-based image processing algorithms. Color image capture equipment is becoming commonplace and recent advances in chip technology make real time color image processing practical.

Shafer's model has shown itself a useful tool for color image understanding [17,18,29,30,31,56]. However, some limitations to the dichromatic theory approach have motivated in part the research presented in this thesis. Some of the major limitations can be summarized as follows:

1. No sensitivity to gray level differences. The theory attributes any linear color variation to the changing illumination geometry of a single material. This limitation does not allow for the finding of material boundaries between objects with collinear matte clusters [29]. In other words, clusters that have similar hues, vary in average intensity, and belong to different objects are not distinguishable.
2. Color information in dark image areas is unreliable. This is similar to what happens within the human eye. At lower light intensities, the eye senses brightness differences, but is unable to make color discriminations.
3. Assumptions about illumination conditions and the object materials do not take into account the possible changes in object reflectance properties and that the illumination color might vary depending on the part of the object being examined.
4. Shafer's model does not take into account more general illumination conditions such as inter-reflection, and shadow-casting between objects, as well as having several light sources that shed differently colored light on the scene.

This thesis will address aspects of the first two limitations. The work is also being motivated from a practical point of view by the following shortcomings of current methods:

1. The inability of state of the art approaches in color image processing to properly segment objects from the background without interference from object shadows and highlights.
2. The inability of existing color similarity measures to evaluate directly color differences based on hue without considering intensity in RGB.
3. The inability of current methods to effectively combine intensity- and hue-based similarity measures when segmenting an image through edge detection, and color clustering approaches.

1.3 Outline

This work is concerned with the extension of image edge detection and image segmentation algorithms into the color image processing realm by examining color-adapted edge detection and image segmentation techniques within different color spaces.

The second chapter describes Shafer's dichromatic reflection model as applied to color images [51]. Various color spaces are then explored in a physics-based context. The third chapter

gives details on the Euclidean distance and the vector angle. It also explores the integration of these two similarity measures into a unified distance measure. The fourth chapter explains the theory behind the edge detection methods explored using the various distance measures. The fifth chapter deals with an aspect of the image segmentation problem; namely color clustering. Results are presented and discussed in the sixth chapter. The thesis concludes with the seventh chapter.

Chapter 2: Color Spaces

2.1 Introduction

Humans perceive color as a result of light in the visible region of the spectrum (i.e., having wavelengths in the region of 400 nm to 700 nm [43]) being projected upon the retina. Therefore, color is the brain's reaction to a specific visual stimulus (i.e., an object reflecting light of a certain wavelength). Although color can be precisely described by measuring its spectral power distribution or SPD (the intensity of the visible electromagnetic radiation at many discrete wavelengths) this leads to a large degree of redundancy. The human retina has only three types of color photoreceptor cone cells each of which responds to incident radiation with a somewhat different spectral response curve. These three broad spectral bands roughly correspond to what humans perceive as red, green and blue light. The rod is the fourth type of photoreceptor cell present in the retina. Rods are effective only at extremely low light levels. The signals from these color sensitive cells (cones), together with those from the rods (sensitive to intensity only), combine in the brain to give "sensations" of different colors [14].

Three numerical components are necessary and sufficient to describe a color given the appropriate use of spectral weighting functions since there are exactly three types of color photoreceptors [61]. This is the concern of the science of colorimetry. In 1931, the Commission Internationale de L'Éclairage (CIE) adopted standard curves for a hypothetical standard observer [42]. These curves specify how an SPD can be transformed into a set of three numbers that specifies a color.

A color space is a method by which we can specify, create and visualize color. Color may be defined by its attributes of brightness, hue and purity for humans. A computer may describe a color using the amounts of red, green and blue phosphor emission required to match a color. A printing press may use the reflectance and absorbance of cyan, magenta and yellow inks on the printing paper to generate a specific color (black ink is also used for gray tones). Thus, a color is usually specified using three co-ordinates, or parameters (except for printing although technically speaking black is not considered a color but the absence of a color). These parameters describe the position of the color within the color space being used. They do not tell us what the color is, that depends on what color space is being used.

This chapter describes the color theory, and color image models or spaces, which are used in this thesis. First, principles of physics-based color image understanding are described using Shafer's Dichromatic Reflection Model [51]. Second, each color space is briefly discussed together with its properties and computational complexity. The computational complexity of

the transformation to obtain each color space will be addressed with respect to the number of addition/subtraction and multiplication/division operations required.

2.2 The Dichromatic Reflection Model

The use of physics can be a very useful tool in studying color. A model for the image formation process based on the product of the illumination E falling on a scene and the reflectance ρ,

$$L(x, y) = E(x, y)\rho(x, y) \tag{1}$$

was first introduced by Stockham [55]. L is the luminance of the formed image at point (x,y).

In this section, a more elaborate reflectance model is described. The Dichromatic Reflection Model first proposed by Shafer [51] is based on physical properties of materials and optics. It is a more generalized form of Stockham's model that takes into account surface geometry and different types of reflection. This theory is further described in [29,30,31] and expanded upon in [17,18,56].

Consider an image of an infinitesimal surface patch of an inhomogeneous dielectric (i.e., non-conducting material) object. To obtain an image of this surface patch the measured sensor values are given by Shafer's [51] model by

$$C = L_b(\lambda, \vec{n}, \vec{s}, \vec{v}) + L_s(\lambda, \vec{n}, \vec{s}, \vec{v})$$

where λ is the wavelength, $\vec{n}$ represents the normal to the surface patch, $\vec{s}$ denotes the direction of the illumination, and $\vec{v}$ is the direction of the viewer. The Dichromatic Reflection Model describes the light which is reflected from a point on a dielectric, non-uniform material as a mixture of the light $L_s(\lambda, \vec{n}, \vec{s}, \vec{v})$ reflected from the material surface or interface (see Figure 1) and the light $L_b(\lambda, \vec{n}, \vec{s}, \vec{v})$ reflected from the material body (see Figure 2).

L_s is called the surface reflection component and generally appears as a highlight or gloss on the object since it is scattered from the surface by a mirror-like reflection that is focused and restricted in angle. L_b is called the body reflection component and exhibits the properties of object shading. Body reflection is defined as the equal scattering of the incident light in all directions when the light exits the material. A fraction of the incident light is absorbed by the material which means only certain wavelengths are reflected.

When an inhomogeneous surface has no interface reflections and only body reflections, it is called a matte or Lambertian surface [61]. Its power spectral distribution corresponds to that of the characteristic wavelength of the surface color. In theory, shadows falling on Lambertian surfaces only change the intensity of the color being reflected thus causing shadows. The power spectral distribution (PSD) of the surface reflection (sometimes called specular reflection) is

generally similar to the power spectral distribution of the illuminating light, whereas, the PSD of the body reflection provides the characteristic object color. Specular reflection is generally called a highlight. For completeness, the definition of inter-reflection is also given. An inter-reflection occurs when light that is reflected from one object falls on another object of a different color. Given that the light reflected from the first object has a similar power spectral distribution to the body reflection of that object, this light will change the color appearance of the object that it falls upon. This thesis does not try to address this complex phenomenon. However, inter-reflections will be described in the results section when appropriate. The reader is directed towards [17] for more information on this topic.

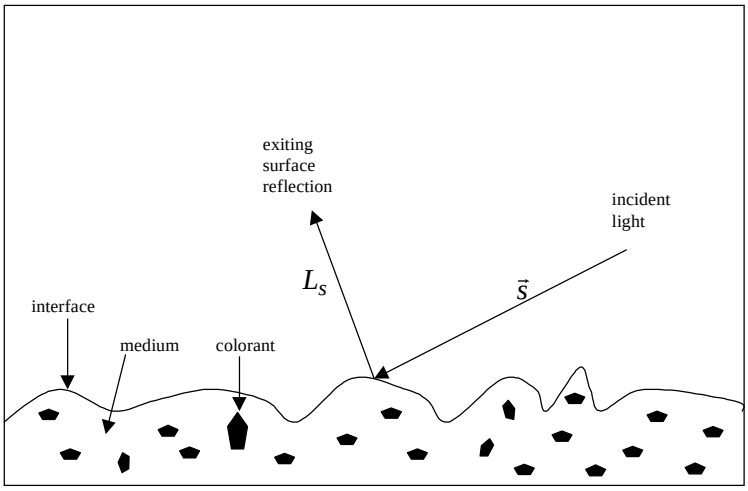


Figure 1: Surface reflection of incident light from dielectric materials

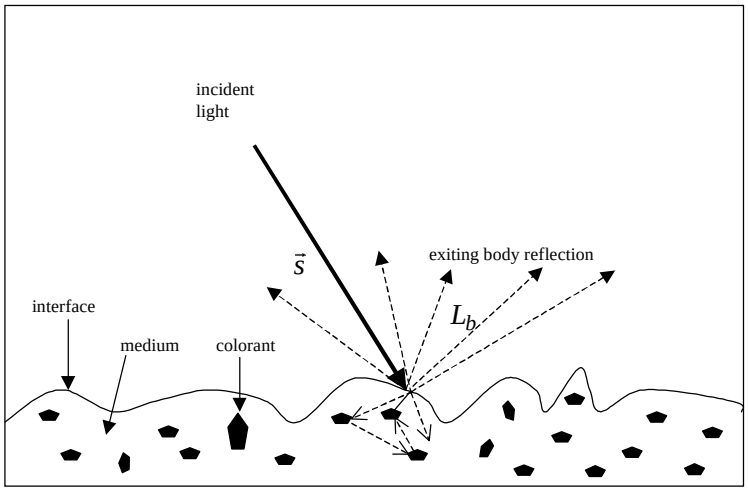


Figure 2: Body reflection of incident light from dielectric materials

In this thesis, only shadows and highlights will be of concern which means that other more complex optical effects such as inter-reflections will still affect the image processing results. To summarize, a shadow on a surface is characterized by a change in illumination intensity (and to some extent saturation) as compared to other regions of the same surface. A highlight is a type of reflection that is characterized by a power spectral distribution similar to the illuminating light. A highlight usually appears as a gloss on an object.

This can be further refined by considering the power spectral distributions of both reflection components, as well as the geometric scale factors. Let $e(\lambda)$ denote the spectral power distribution of the incident light. Let the spectral sensitivities of the red, green and blue (or RGB) sensors be $f_R(\lambda)$, $f_G(\lambda)$, and $f_B(\lambda)$, respectively. Consider the following response for sensor C [17],

$$C = m_b(\vec{n},\vec{s})\int_\lambda f_C(\lambda)e(\lambda)c_b(\lambda)d\lambda + m_s(\vec{n},\vec{s},\vec{v})\int_\lambda f_C(\lambda)e(\lambda)c_s(\lambda)d\lambda$$

for $C \in \{R,G,B\}$. The spectral power distribution of the light reflected from the surface (also known as the surface albedo) is denoted by $c_s(\lambda)$ while $c_b(\lambda)$ represents the spectral power distribution of the light reflected from the body of the object (also known as Fresnel reflectance [17]). The $m_b(\vec{n},\vec{s})$ and $m_s(\vec{n},\vec{s},\vec{v})$ represent respectively the geometric scale factors for the body and surface reflections.

Since the reflection from the body is distributed among many angles and the reflection from the interface is confined to a small angle, highlights on a surface appear much brighter than diffuse reflection from the body [61]. The differences in geometrical distributions of these reflected rays indicate that highlights can only be seen from certain angles. This is commonly experienced by all humans every day.

The reflected light can be described as a linear combination of the two vectors $c_b(\lambda)$ and $c_s(\lambda)$ in the infinite-dimensional vector space of spectral power distributions where each wavelength defines an independent dimension. However, for the purposes of this thesis, the power spectral distributions at the primary red, green and blue wavelengths will be considered. The spectral sensitivity of the sensors of each camera is slightly different. For example, in the human eye the spectral sensitivity of blue cones peaks at 420 nm, that of green cones peaks at 534 nm and that of red cones peaks at 564 nm, while the spectral sensitivity of rods (sensitive to light brightness) peaks at 496 nm [4].

Assuming that $c_s(\lambda)$ has a constant value independent of the incident light wavelength and that the illumination source $e(\lambda)$ is what humans see as white light (i.e., the energy density is equal for all wavelengths in the visible spectrum from 400 nm to 700 nm), then

$$c_s = c_s(\lambda)$$
$$e = e(\lambda)$$

Next, let k_C be the compact formulation of the dependency on surface albedo and sensors described by

$$k_C = \int_\lambda f_C(\lambda) c_b(\lambda) d\lambda$$

The sensor response can be now rewritten as

$$C_w = m_b(\vec{n}, \vec{s}) k_C + e m_s(\vec{n}, \vec{s}, \vec{v}) c_s \int_\lambda f_C(\lambda) d\lambda$$

for $C_w \in \{R_w, G_w, B_w\}$ representing the RGB sensor response under the assumption of white light.

Assuming the light source is white light, Gevers and Smeulders [17] state that

$$f = \int_\lambda f_R(\lambda) d\lambda = \int_\lambda f_G(\lambda) d\lambda = \int_\lambda f_B(\lambda) d\lambda \tag{2}$$

However, in practice this assumption seems to hold only for colors with higher intensities or colors with high saturation (i.e., relatively pure colors). This is perhaps because the RGB bands do not necessarily have approximately the same widths. In other words, this assumption fails at low intensity levels. This is a related phenomenon to the occurrence of shadows on matte surfaces described above and will be addressed when discussing results from color spaces claiming to be invariant to certain physical occurrences such as shadows and highlights.

Finally, the reflection from a non-uniform dielectric material under white illumination can be written as

$$C_w = e m_b(\vec{n}, \vec{s}) k_C + e m_s(\vec{n}, \vec{s}, \vec{v}) c_s f \tag{3}$$

The model is thus a useful tool for describing the light that is reflected from an infinitesimal patch (or in discrete terms a point) of an object as a mixture of two distinct power spectral distributions, $c_b(\lambda)$ and $c_s(\lambda)$. Each of the PSD's is scaled according to the properties of body and surface reflection.

In the next section, color image models will be discussed together with their properties with respect to Shafer's Dichromatic Reflection Model. All theoretical proofs will be derived for Lambertian (i.e., matte) surfaces and metallic or specular surfaces will not be of concern.

2.3 Color Image Models

Color is a phenomenon of perception, not an objective component or characteristic of a substance. Color is an aspect of vision; it is a psychophysical response consisting of the physical reaction of the eye and the automatic interpretive response of the brain to wavelength characteristics of light above a certain brightness level. At lower light intensity levels, the eye senses brightness differences, but is unable to make color discriminations.

There are many different ways of representing color. The most common way in computer graphics is to use a triplet of intensity values. Any unique combination of the three values yields a distinct color. The three dimensional space which describes the distribution of physical colors is called a color space.

Color vectors in each of these spaces differ from one another. This difference means that two colors in one space being separated by one distance value would be separated by a different distance value in another space. A color represented in one space can be changed to another spatial representation by performing some linear or non-linear transformation. As mentioned above, the RGB space is the physical sensor-based color space. It is from this space that all other color spaces are derived. It is still used more than any other space today in image processing applications due to its direct relationship with the physical world. There are many other spaces including CMY [19], YIQ, YUV, HSI [19] or its generalized form HSV [33], CIELAB [33], CIELUV, rgb (known as the normalized RGB space) [19], $c_1c_2c_3$ [17], $l_1l_2l_3$ [17], YC_bC_r [41], and the Angle Space [70]. Figure 3 shows the relationships between the color spaces. It also gives the nature of the transform required to go from one space to another.

The concentration of research on the RGB color space is logical since it avoids the preprocessing step of transforming the captured image into another color space. In the past, only YUV and YIQ transformations seemed to be fast enough to allow processing of images in real time. Today, transformations to such spaces as rgb, $l_1l_2l_3$, and $h_1h_2h_3$ are also very fast, but they present other problems such as random behavior at low intensities. Such transformations as CIELAB, CIELUV and various flavors of HSI are nonlinear in nature and are very computationally intensive. It is, therefore, desirable to find fast algorithms for use in the RGB space for real-time applications such as check processing [65].

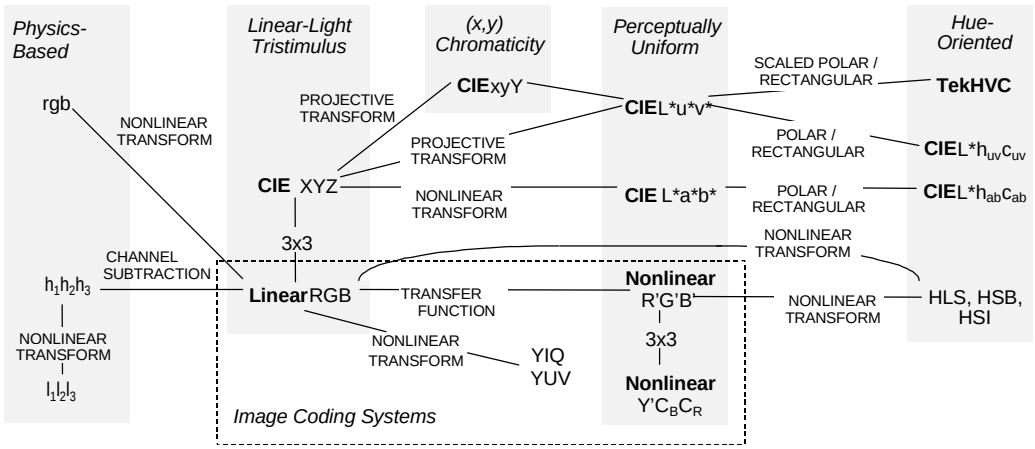


Figure 3: Color Systems (based on [41])

In this thesis, the following color spaces will be examined for either edge detection or color clustering applications: RGB, XYZ, CIELAB, CIELUV, rgb (i.e., normalized RGB), $l_1l_2l_3$, and $h_1h_2h_3$. The first four have been chosen due to their extended use in the literature. The other three spaces, rgb, h_1h_2h, and $l_1l_2l_3$ are combinations of RGB values that create spaces with various invariant properties with respect to the Dichromatic Reflectance Model. The transformations from the RGB space to this subset of spaces can be found in the sections below and in [33,44,19]. A detailed description of the color spaces used in this thesis follows.

It is necessary to examine several color spaces with respect to the same edge detection and image segmentation algorithms since each coordinate system represents data differently. Some spaces such as CIELAB and CIELUV have been especially designed to be perceptually correct; that is, the Euclidean distance is supposed to indicate perceptual color differences accurately, while for example the RGB space is based on the physical (i.e., wavelength) representation of the three primary colors red, green and blue. It would be of great importance to find out whether the algorithms proposed in this thesis function better (given the quantitative evaluations techniques used in this thesis) in the perceptually correct spaces rather than in the RGB (or some other) space for several reasons:

1. Computational speed
2. Better understanding of how color representation affects image processing results
3. Better understanding of the interaction between the color distance measures and the color space in which they are used

The computation in the RGB space will serve as a baseline for the comparison with other color spaces given its widely known properties and its popularity. The CIELAB and CIELUV spaces will allow for the evaluation of perceptually correct spaces. The rgb, h_1h_2h, and $l_1l_2l_3$ spaces will give insights into the processing of images in physics-based color representations.

The computational complexity of all color space transformations with respect to the number of additions/subtractions (A/S) and multiplications/divisions (M/D) will be given. All operations are assumed to be floating point.

2.3.1 RGB

The RGB (Red, Green, Blue) space is used most frequently in computer graphics and image processing applications. A color in this space is represented by a triplet of values typically between zero and one and is usually scaled by 255 for an 8-bit representation. Each color can be broken down into its relative intensity in the three primaries corresponding to the spectral response of one of the three types of cones present in the human eye: red, green and blue. The space is easily represented as a three dimensional cube where each axis represents the strength of the color in one of the three primaries (see Figure 4).

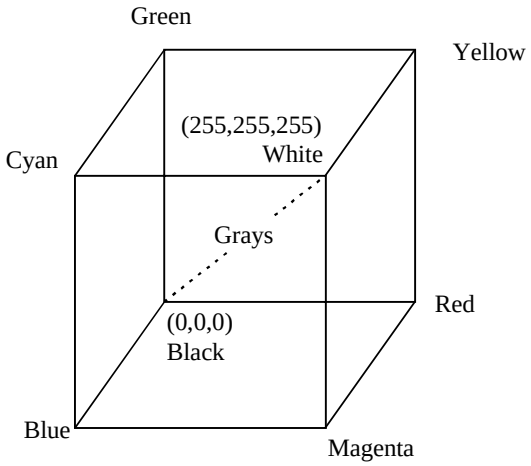


Figure 4: The RGB color space

The quantization of each component into an 8-bit value introduces some round-off effect. With respect to human sensitivity, experimental studies indicate that a 2% change in graylevel contrast on a video screen is just noticeable [28]. Thus, if the quantization is uniform the contrast scale needs about 50 levels or 6 bits. Therefore, assuming the human eye should be able to distinguish between approximately 50^3 colors and that each plane of the RGB space is quantized into an 8-bit value (i.e., 256 levels), quantization should not be an issue in this thesis.

The dichromatic reflection theory states that the body reflection term from equation 3 depends on the sensors and surface reflective power (i.e., k_C), as well as the illumination intensity and the object geometry. For a matte surface region containing a variety of surface normals (e.g., due to changing intensity or because of shadows), the set of measured colors will generate an elongated cluster. This cluster will be composed of a broad range of RGB pixels although the intrinsic color or hue at each pixel should be the same. Furthermore, color highlights are only related to the color of the light source and not to the color of the surface (under the assumption

that $c_s(\lambda)$ is constant for all values of λ). This means that the sensor-measured values of a shiny surface under a white light can be represented by gray values within the RGB space. The extent of the streak will depend on the reflectiveness of the object surface. A rough surface will generate a small cluster while for very shiny objects the cluster will extend along much of the gray axis [17]. This also leads to a large number of RGB values.

Therefore, by applying Shafer's reflection model, one easily notices that the color representation in the RGB space is sensitive to the viewing direction, object surface orientation, highlights (i.e., gloss, cf. Section 2.2), illumination direction, illumination intensity, illumination color, and inter-reflection.

To obtain this representation no transformations need to be applied. Therefore, the number of computations is zero.

2.3.2 XYZ

The XYZ color space was developed by CIE as an alternative to RGB [33]. Since it is impossible to choose three real primaries such that all possible colors can be matched with additive mixtures of those primaries, three imaginary primaries were created when the CIE system was specified in 1931 [41]. One of the characteristics of this system is that the tristimulus values *X*, *Y*, and *Z*, are always positive for all real color stimuli [41]. Poyton states that there were several reasons for the adoption of imaginary primaries [42]:

1. It was necessary to devise *X*, *Y*, and *Z* such that they would be positive for all possible real stimuli.
2. The coefficients were chosen such that the *Y* tristimulus value was directly proportional to the luminance of the additive mixture.
3. The coefficients were chosen such that *X*=*Y*=*Z* for a match to a stimulus that has equal luminance at each wavelength.

The XYZ color space can be obtained by transforming the RGB space by

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.607 & 0.174 & 0.200 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.116 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4}$$

Since the XYZ space is just a linear translation of the RGB space, the conclusions drawn from using the dichromatic reflection model to analyze the RGB space will also apply. Therefore, the color representation in the XYZ space is sensitive to the viewing direction, object surface orientation, highlights, illumination direction, illumination intensity, illumination color, and inter-reflection.

To obtain the XYZ color space nine M/D's and six A/S's are required per point.

2.3.3 CIELAB

The CIELAB cube root color coordinate system was developed to give a simple measure of color in agreement with the Munsell color system. Professor Albert H. Munsell, both an artist of distinction and a gifted teacher of art, developed the first widely accepted color order system to make the description of color accurate and convenient and to aid in the teaching of color. The Munsell color order system has gained international acceptance and has served as the foundation for other color order systems [41].

There are perhaps two problems with the specification of colors in terms of tristimulus values and chromaticity space. First, the RGB space is not easily interpreted in terms of the psychophysical dimensions of color perception such as brightness, hue, and purity. Second, the XYZ system and the associated chromaticity diagrams are not perceptually uniform which means that differences in color are not equally perceived in the whole CIELAB space. For example, the difference between colors A and B might be perceptually smaller than the distance between colors A and C even though color vectors A and C are closer in terms of the Euclidean distance. The second of these points is a problem if we wish to estimate the magnitude of the difference between two color stimuli. The need for a uniform color space led to a number of nonlinear transformations of the CIE 1931 XYZ space and finally resulted in the specification of one of these transformations as the CIE 1976 L*a*b* or CIELAB color space [41].

The CIELAB space has been designed to be a perceptually uniform space. A system is perceptually uniform if a small perturbation to a component value is approximately equally perceptible across the range of that value [14]. For example, the volume control on a radio could be designed to be perceptually uniform: rotating the knob twenty degrees anywhere across the range will produce approximately the same perceptual increment in volume. Therefore, the logarithmic nature of human perception of sound would place the perceptual difference of the control at the bottom of its range if the control were physically linear [42].

In CIELAB, the L-axis is known as the lightness and extends from 0 (black) to 100 (white). The other two coordinates A and B represent redness-greenness and yellowness-blueness respectively [44]. Samples for which A=B=0 are achromatic. Therefore, the L-axis represents the achromatic scale of grays from black to white. The non-linear relationships for L, A, and B are intended to mimic the logarithmic response of the eye [14].

As stated above, the CIELAB space is a perceptually correct space. This means that a perceptual difference between two points in the CIELAB space can be represented closely by the Euclidean distance (square norm) measure. The XYZ-to-CIELAB transformation is shown in Equation 5:

$$L = \begin{cases} 25\left[100\frac{Y}{Y_0}\right]^{1/3} - 16 & if\ \frac{Y}{Y_0} \geq 0.008856 \\ 903.3\frac{Y}{Y_0} & otherwise \end{cases} \quad (5)$$

$$A = 500\left[\left[\frac{X}{X_0}\right]^{1/3} - \left[\frac{Y}{Y_0}\right]^{1/3}\right]$$

$$B = 200\left[\left[\frac{X}{X_0}\right]^{1/3} - \left[\frac{Z}{Z_0}\right]^{1/3}\right]$$

where X_0, Y_0, Z_0 are the tristimulus values for the reference white (i.e., RGB=255,255,255). For the purpose of this thesis, it is assumed that the light source is white in all cases.

Given that CIELAB space is a nonlinear transformation of the XYZ space, the dichromatic reflection model states that the color representation in this space is sensitive to the viewing direction, object surface orientation, highlights, illumination direction, illumination intensity, and inter-reflection. In theory, it is invariant to the illumination color since the XYZ values are normalized with respect to a white reference. However, in practice, it is only possible to obtain this property when the capturing device is constantly in a controlled environment such as check processing or industrial part inspection.

For this transformation, there are 18 M/D's and 9 A/S's, as well as 11 M/D's and 6 A/S's for the computational load of carrying out each of the three different cubic roots assuming that only the first three terms are used in the Taylor series for limited precision. This is reasonable given the coarser precision of the image quantization. This will be also assumed for all other series approximations when calculating computational complexity. Together with the RGB-to-XYZ transformation, the total number of operations per pixel is 60 M/D's and 33 A/S's.

2.3.4 CIELUV

The CIELUV color coordinate system has evolved from the CIELAB and UVW spaces [44]. The CIELUV space has also been designed to be a perceptually correct space (it also improves the perceptual nonuniformity ratio of the XYZ color model to 6:1 [42]); i.e., the Euclidean distance between two points indicates more or less the perceptual difference between them. Furthermore, the CIELUV space exhibits differences in color typical of those in the Munsell book of color, whereas the CIELAB space was intended to exhibit color differences greater than the JND threshold, but smaller than those in the Munsell book of color [28]. The XYZ-to-CIELUV transformation is shown in Equation 6 below. In CIELUV, L is a measure of intensity; however, there is no analogous relationship between U and V and the RGB components as there is for the A and B planes in the CIELAB space. The nonlinear relationship

for L is intended to mimic the logarithmic response of the eye [14]. The transformation is defined as,

$$L = \begin{cases} 25\left[100\dfrac{Y}{Y_0}\right]^{1/3} - 16 & \dfrac{Y}{Y_0} \geq 0.008856 \\ 903.3\dfrac{Y}{Y_0} & \dfrac{Y}{Y_0} < 0.008856 \end{cases} \tag{6}$$

$$U = 13L\left(u' - u'_0\right)$$
$$V = 13L\left(v' - v'_0\right)$$

where u_o and v_o are obtained by the substitution of the tristimulus values X_0, Y_0, Z_0 for the reference white and u' and v' are defined as

$$u' = \frac{4X}{X + 15Y + 3Z} \tag{7}$$

$$v' = \frac{9Y}{X + 15Y + 3Z} \tag{8}$$

The analysis of the CIELUV space with the dichromatic reflection model is worth exploring.

Consider the calculation of the U component:

$$U = 13L\left(u' - u'_0\right)$$

where u' is defined in equation 7.

Applying the dichromatic reflection model for matte or Lambertian surfaces (i.e., only the body reflection term is considered since such surfaces do not shine) to equation 7 (without loss of generality to equation 8) yields

$$u' = \frac{2.428 \cdot em_b(\vec{n},\vec{s})k_R + 0.696 \cdot em_b(\vec{n},\vec{s})k_G + 0.800 \cdot em_b(\vec{n},\vec{s})k_B}{5.092 \cdot em_b(\vec{n},\vec{s})k_R + 9.177 \cdot em_b(\vec{n},\vec{s})k_G + 5.258 \cdot em_b(\vec{n},\vec{s})k_B}$$

Factoring out the common terms, the result is

$$u' = \frac{2.428 \cdot k_R + 0.696 \cdot k_G + 0.800 \cdot k_B}{5.092 \cdot k_R + 9.177 \cdot k_G + 5.258 \cdot k_B}$$

which is a ratio which depends only on sensors and surface albedo. Therefore, the representation of the color in the CIELUV space is invariant with respect to illumination direction and intensity, as well as viewing direction and object surface orientation (since the geometry term has been canceled out). This would mean that in the CIELUV space the effect of shadows should be much reduced with respect to the RGB space. However, this space varies with respect to highlights as can be easily verified by analyzing the surface reflection term of equation 3. Therefore, highlights should appear in CIELUV just as well as they appear in RGB.

The color representation in this space is sensitive to the highlights, and inter-reflection. Since the colors are normalized with respect to a white reference (i.e., to the values corresponding to a white color under the illuminating light), it would be reasonable to say that color represented by CIELUV is invariant to the illumination color. However, like CIELAB where the white reference is obtained a priori, this conclusion only applies to well controlled lighting environments.

For this transformation, there are at least 14 M/D's and 7 A/S's, as well as 11 M/D's and 6 A/S's for the computational load of carrying out one cubic root (cf. see Section 2.3.3). Together with the RGB-to-XYZ transformation, the total is 34 M/D's and 19 A/S's.

2.3.5 rgb

The normalized RGB space, or rgb, has been used in the literature for several decades [19]. The following transformation needs to be performed to obtain this space:

$$r = \frac{R}{R+G+B} \qquad (9)$$
$$g = \frac{G}{R+G+B}$$
$$b = \frac{B}{R+G+B}$$

It follows that all the values need to add up to one.

When Shafer's model is applied to the rgb space [17], the same conclusions as were made for CIELUV can be made. Namely, for matte objects the color representation in the rgb space is invariant with respect to illumination direction and intensity, as well as the viewing direction and surface orientation. This space is still sensitive to highlights, and inter-reflection.

For this transformation, there are 3 M/D's and 2 A/S's.

2.3.6 HSI

For completeness the HSI space is defined, however it will not be used in this thesis. As stated previously, the HSI (Hue, Saturation, and Intensity) model [35] can be said to follow more closely the human perception of color qualities. Hue (H) is the color as described by wavelength – for example, the distinction between red and blue. Hue represents the fundamental or dominant color. Saturation (S) represents the amount of a color present, where pastel shades (e.g. pink) have low saturation values while pure spectral colors (e.g. red) are completely saturated. The intensity (I) represents the overall brightness or the amount of light. It is independent of color and is a linear value. It is measured as an angle on a color circle with the three primary colors spaced 120° apart. The first two values specify the chromaticity of a color point. Figure 5 shows the relationship between the HSI and RGB spaces.

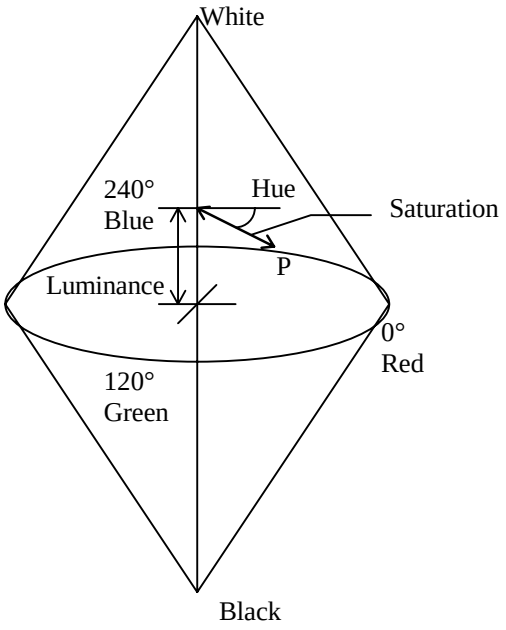


Figure 5: HSI Color Space

It is noted that the HSI color space is one of several spaces that can be derived from the General Hue, Luminance and Saturation (GHLS) space. There are other slightly different interpretations of hue, luminance and saturation. There is a general transformation from RGB to GHLS [33]. By setting certain parameters in this transformation, one can specify any of the transformations from the RGB space to an HSI, HLS or HSV [33] (Hue, Saturation and Value) representation.

The HSI definition is given in [19]:

$$\cos\theta = \frac{(R-G)+(R-B)}{2\cdot\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}} \tag{10}$$

$$H = \begin{cases} \theta & \text{if } B < G \\ 2\pi - \theta & \text{otherwise} \end{cases}$$

$$S = 1 - \frac{3\cdot\min(R,G,B)}{R+G+B}$$

$$I = \frac{1}{3}(R+G+B)$$

The plane I=1 (for normalized R, G, B values) defines the hue triangle (with vertices at the extremities of the Red, Green and Blue axes) from which these equations have been derived. Shafer's model was applied to the different components of the HSI space [17]. Several conclusions can be drawn from this analysis. First, that the I plane is sensitive to the various physical effects of light. Second, the saturation plane representation is invariant with respect to illumination direction and intensity as well as the viewing direction and surface orientation.

Finally, the hue plane has several invariant properties with respect to highlights, viewing direction, surface orientation, as well as to illumination direction and intensity.

However, although the theoretical results seem to present a good case for HSI, there are many problems. Poynton gives several arguments against the use of HLS-based (e.g. HSI) color systems [43]:

1. For example, he indicates that the HLS spaces do not refer to the lightness perception by humans when computing the lightness or intensity components as spaces such as XYZ or CIELUV do.
2. There is no reference to the linearity or nonlinearity of the underlying RGB color space. For example, the computation for the intensity presented in Equation 10 would indicate yellow to be about six times more intense than blue at the same lightness level of 50 out of a 0-100 range (as determined from the Y component of XYZ). That is, the intensity component in HSI is not represented by the weighed average of the RGB components as perceived by humans.
3. Nearly all formulations of HLS introduce visible discontinuities in color space due to different computations around 60 degree segments of the hue circle.
4. The ubiquitous formulations are based on RGB components whose chromaticities and white point are unspecified which indicates that the HLS-type systems do not convey accurate color information.

For these reasons, the HSI space will not be used in this thesis.

The RGB to HSI transformation is very computationally intense: eight additions/subtractions and seven multiplications/divisions, as well as 11 M/D's and 6 A/S's for a square root operator (based again on a three term Taylor series expansion) per point for a total of 18 M/D's and 14 A/S's.

2.3.7 $l_1l_2l_3$

Gevers and Smeulders observed that the measured colors of a uniformly colored region must be on the triangular color plane in the RGB space spanned by the body and surface reflection components [17]. Therefore, just as hue (in the HSI model) is a function of the angle between the main diagonal and the color point in RGB, any expression defining colors on the same linear triangular plane will have similar properties to hue. The $l_1l_2l_3$ space was introduced as a space that uniquely determines the direction of the triangular color in the RGB space [17]. The space is described as follows:

$$l_1 = \frac{(R-G)^2}{(R-G)^2 + (R-B)^2 + (G-B)^2} \qquad (11)$$
$$l_2 = \frac{(R-B)^2}{(R-G)^2 + (R-B)^2 + (G-B)^2}$$
$$l_3 = \frac{(G-B)^2}{(R-G)^2 + (R-B)^2 + (G-B)^2}$$

Absolute values can be used instead of squaring in order to increase algorithm speed [18]. Naturally, applying the dichromatic reflection theory one may observe that this space is invariant to highlights, viewing direction, surface orientation, as well as to illumination direction and intensity.

For this transformation, there are 6 M/D's and 5 A/S's. If absolute values are used instead of squaring, there will only be 3 M/D's and 5 A/S's.

2.3.8 $h_1h_2h_3$

The rgb space has show itself to be invariant to illumination intensity. The $l_1l_2l_3$ space has shown invariance with respect to highlights and illumination intensity. A new space will now be introduced which is only invariant with respect to highlights. This space will be preliminarily called the $h_1h_2h_3$ space and will be defined as follows:

$$h_1 = R - G \qquad (12)$$
$$h_2 = G - B$$
$$h_3 = B - R$$

There are many color spaces already and introducing another one would seem redundant. However, there are several motivating factors for introducing the $h_1h_2h_3$ space. First, although difference-based color spaces exist, there is no color space that is solely based on channel differences. Second, the space has the property of having very low values when pixel colors in the RGB space lie close to the R=G=B line (i.e., gray values). In this way, the intensity component is easily removed. Third, Shafer's model shows that this space is invariant to highlights, a useful feature in image understanding when observing shiny surfaces. This completes the gamut of spaces based on Shafer's model that exhibit different kinds of invariant physics-based properties.

By applying the dichromatic reflection model, it is easily seen that the new space only depends on the body reflection term for each of the spectral components. Consider the h_1 term (without loss of generality):

$$h_1 = em_b(\vec{n},\vec{s})k_R + em_s(\vec{n},\vec{s},\vec{v})c_s f - \left(em_b(\vec{n},\vec{s})k_G + em_s(\vec{n},\vec{s},\vec{v})c_s f\right) \qquad (13)$$
$$h_1 = em_b(\vec{n},\vec{s})\left[k_R - k_G\right]$$

This means that the terms in this new space are sensitive to viewing direction, surface orientation, as well as to illumination direction and intensity. Although, this space is invariant to highlights in theory, this does not seem to be the case in practice. This will become important when discussing the distance measures in the next section. In addition, an immediate practical consequence of this transformation is that any gray-level area will have coordinates very close to zero irrespective of the intensity level. That is, the magnitude of a gray valued color vector in $h_1h_2h_3$ space will be near zero.

For this transformation, there are only three additions/subtractions.

2.4 Summary

Table 1 shows a summary of the characteristics of the color spaces discussed in this chapter. Most of the color spaces have an important individual feature or a unique combination of features from other spaces. Some color spaces such as RGB, XYZ, CIELAB, and CIELUV are provided for comparison purposes with previous work.

All the spaces presented in Table 1 will be used in the thesis except for the HSI space for the reasons outlined above. From this table, several hypotheses can be made assuming the Dichromatic Reflectance model is valid. First, the RGB, XYZ, CIELAB and $h_1h_2h_3$ spaces should produce worst results when using the Euclidean distance (cf. definition in Chapter 3) given the low number of invariances indicated. The best results should be obtained from the CIELUV, rgb, and $l_1l_2l_3$ spaces given the high number of invariances indicated. However, given the perceptually uniform nature of CIELAB, the image edge detection and region segmentation results should appear better than results in non-perceptually uniform spaces.

The use of other color similarity measures will be discussed in the next chapter, as these will introduce certain changes to the color space properties. That is the color space properties depend very much on the color similarity measure that is being used to compute color differences. The properties discussed in this chapter can only apply to the use of the Euclidean distance.

Finally, the CIELAB and CIELUV space transformations will be the most computationally intensive whereas processing in RGB will have no overhead associated with color space transformations.

Color Space	Conversion Complexity From RGB		viewing direction	surface orientation	illumination direction	illumination intensity	illumination color	highlights	perceptual uniformity	unique features
	M/D's	A/S's								
RGB	0	0	+	+	+	+	+	+	+	Sensor-based; easily understood from a physics perspective
XYZ	9	6	+	+	+	+	+	+	+	Y channel represents luminance
CIELAB	60	33	+	+	+	+	+/-	+	-	Perceptually uniform
CIELUV	34	19	-	-	-	-	+/-	+	-	Perceptually uniform
rgb	3	2	-	-	-	-	+	+	+	Random values for small values of R, G and B
HSI	18	14	-	-	-	-	+	-	+	Very intuitive; I channel represents luminance; Hue channel shows color of pixels
$\mathbf{l_1 l_2 l_3}$	6	5	-	-	-	-	+	-	+	Random values for small values of R, G and B
$\mathbf{h_1 h_2 h_3}$	0	3	+	+	+	+	+	-	+	Channel subtraction

Table 1: Color space characteristics ('-' means "invariant to", '+' means "sensitive to") based on the Dichromatic Reflectance Model

Chapter 3: Color Similarity Measures

3.1 Introduction

There are different color similarity measures used in the literature today depending on the color space. For example, the Euclidean distance (cf. definition in Section 3.3) is used on RGB [70], CIELUV [49,50], CIELAB [68] and even on a non-standard hybrid multidimensional space made up of RGB, XYZ, and LAB spaces [69]. Variations on the Euclidean distance have been used in HCI [8], and the Angle space [70]. Some researchers have also applied intensity-based difference measures on individual color planes in the RGB and HSI spaces and then combined the results [7,24]. Extracting useful information for color edge detection or image segmentation directly from the RGB space is not well defined. However, given that images are easily available in the RGB space, most research has been done in that domain. It seems that using a perceptually correct space or a space with a hue-based component is necessary to process the information in these images with greater accuracy.

Furthermore, there has been a lot of successful development in the area of color space transformations to derive hue-based or perceptually correct spaces for the past twenty five years [17,28,33,42]. This has resulted in several very useful color representations; however, at this time no RGB hue-based measure has been introduced in the literature. Recently, the vector angle has been applied to the RGB space to address this problem [12]. Several distance measures are described in [1] including the Euclidean distance and a different form of the vector angle measure. Other similarity measures such as the Canberra distance, the Czekanowski coefficient, as well as the generalized Minkowski metric are also described. However, since they have been found to perform poorly in a color-based image similarity study for an image retrieval application [1], they will not be considered in this thesis.

A much more complex way to calculate color differences is described in [28]. It consists of describing small differences in color on observations of just noticeable differences (JNDs) from one color to another. A unit JND is described by an ellipsoid with a set of nine parameters (three for each plane). The JND ellipsoids would be of uniform size if these coefficients were constant throughout the particular color space. This would mean that the color space could be reduced to a Euclidean tristimulus space where the color differences between any two colors would become proportional to the length of the straight line joining them [28]. However, these coefficients exhibit large variations in RGB which means that the size and orientation of each of the JND ellipsoids varies considerably. Because of this the distance between two arbitrary colors is described by a geodesic (i.e., the minimal distance chain of ellipsoids lying along a

curve joining the two colors). For example, in the CIELUV space the geodesics between the primary colors red, green and blue are nearly straight lines in the chromacity (u,v) plane, whereas the geodesic curves between other colors are generally curved [28]. Because of the complexity of computing geodesics, this thesis will not discuss this topic. The reader is encouraged to consult [28] for further information and references.

There has been some research done on combining distance measures in color image processing [1,66]. Some researchers have even chosen an entirely different route: that of coming up with the optimal set of three color planes from a set of multiple color spaces [60]. However, in this thesis, the focus will be on examining the currently available color spaces and coming up with the best combination of color space and distance measures for quantifying color differences for the purposes of color edge detection and image segmentation. A mixture of the Euclidean distance and the vector angle measures are also examined in the RGB color space for color edge detection applications. Various algorithms have been examined using different color spaces [9,34,40]; however, there has not been an evaluation of color distance measures on the various color spaces.

Two areas are of interest:

1. To devise similarity measures with invariant properties in the RGB space. After applying each of those distance measures, a cumulative distance measure is calculated based on a weighted combination of the individual measures.

2. To devise special color spaces with invariant properties. Next, the image segmentation methods using the Euclidean distance can be applied to each of those spaces. Finally, an algorithm combines the result.

The first point names an emerging field, but results are scarce [1]. Recently, the work of Carron and Lambert in the HCI space [7] was applied to the problem of edge detection in the RGB space using a combination of the Euclidean distance and the vector angle [66]. The work in this thesis will demonstrate how one can use only RGB or one of the other color spaces with specifically designed distance measures to segment color images without the need to design any new color spaces.

The second point represents an extension of using the HSI space in color image understanding with some work having already been done [17,18]. There has been some research done in this field using HSI [7]. This methodology will not be explored in this thesis.

In this chapter, first an artificial image is described that will be used for comparing the similarity measures for various color spaces. Next, the Euclidean distance measure is discussed. Then, the vector angle measure is explained in detail. Third, addition-based combinations of the two similarity measures are introduced and discussed. The combined computational complexity

of using a certain color space and the various distance measures is addressed in detail. Particular attention is paid to the properties resulting from the combination of each similarity measure and the different color spaces.

3.2 Example

To examine the empirical effects of using Euclidean distance in the RGB space, a test image was introduced in [12]. Please refer to the example image in Figure 6 and the vector representation of the four colors in this image in Figure 7. This image was chosen so that colors C and D, as well as colors A and B have the similar hues. In addition, colors B and C are of similar intensity as are colors A and D.

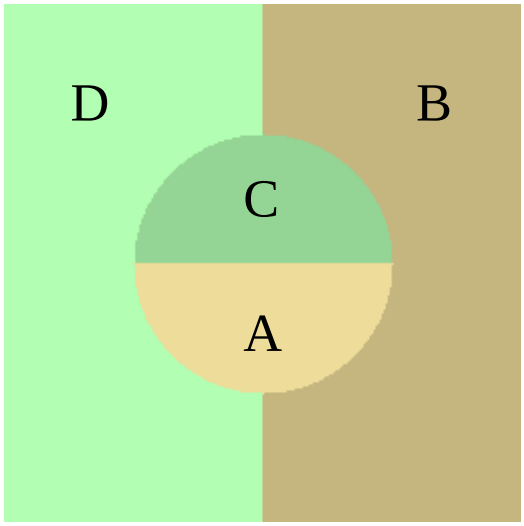


Figure 6: Artificial test image with four colors

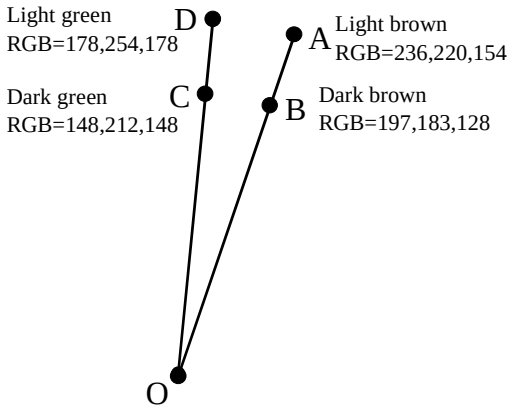


Figure 7: Vector geometry between the four colors of Figure 6

The purpose of this example is to show that the Euclidean distance is not necessarily a good color discriminator for all of the spaces considered. A desired outcome is to show that in RGB the vector angle can distinguish colors based on hue where the Euclidean distance fails to do so.

3.3 Euclidean Distance

The Euclidean distance (D_E) measure is usually used to compute distance in N-dimensional vector space. It is defined as

$$D_E(\vec{v}_1,\vec{v}_2) = \left\|\vec{v}_1 - \vec{v}_2\right\| \tag{14}$$

where $\|\bullet\|$ is the L_2 vector norm. For a three-plane color space the distance calculated is

$$D_E(\vec{v}_1,\vec{v}_2) = \sqrt{(v_{1,1} - v_{2,1})^2 + (v_{1,2} - v_{2,2})^2 + (v_{1,3} - v_{2,3})^2} \tag{15}$$

where $\vec{v}_1 = [v_{1,1}\ v_{1,2}\ v_{1,3}]^T$ is a color triplet.

The computational expense of using the Euclidean distance is five additions/subtractions and three multiplications, as well as 11 M/D's and 6 A/S's for one square root operator (assuming a three term Taylor expansion) per comparison for a total of 14 M/D's and 11 A/S's. However, the total computational load will vary depending on the number of calculations required for each of the image segmentation algorithms.

The results obtained using Shafer's model in the previous chapter apply of course when the Euclidean distance is used and, therefore will not be discussed here. The results of comparing the color difference calculation for the artificial image example presented in Section 3.2 will be discussed with respect to the different color spaces used.

3.3.1 RGB

In RGB, the Euclidean distance represents at once differences in intensity, hue and saturation. It is not clear in what proportion each of these differences is represented. It seems that intensity counts a great deal when calculating a color difference. That is, the magnitude of a vector (i.e., an elongation or shortening of the vector) will produce differences in the calculation of the distance. This illustrates the effect of illumination variations on the resulting color pixel value and color differences.

In RGB coordinates, two pixel values from areas of the same reflectivity characteristics, but under different illumination intensities (assuming equal color balance in the illumination) are related by [12]

$$\vec{v}_2 = \alpha \cdot \vec{v}_1 \tag{16}$$

where α is a scalar such that $\alpha > 0$ and $\alpha \neq 1$. Given that the two pixels in RGB come from areas with the same hue (i.e., intrinsic color as described by wavelength), the Euclidean distance calculation would not be zero due to the variation in illumination; i.e.,

$$D_E(\vec{v}_1, \vec{v}_2) \neq 0 \tag{17}$$

Therefore, the distance between two colors would depend very much on intensity. This is a undesirable if trying to compute distances between two different colors that have similar intensities. However, it is a desirable feature when trying to determine differences between two colors of the same hue.

Typically, the goal in image analysis is to determine characteristics about the underlying physical properties of the scene being imaged such as differences in material from one object to the next. These are inferred from the reflectance properties of the objects present in the scene. Therefore, the way light is reflected from the various objects conveys the information about the scene. Variations in the illumination can be considered as noise most of the time (in practice the only time this is not true is when two objects of the same hue but of different saturation values overlap spatially).

In the above example, it may be the case that color samples *A* and *B* come from regions with exactly the same reflectance properties, but region *A* has 20% more incident illumination. For image segmentation, there should be no difference between *A* and *B* or between *C* and *D*; however, using the Euclidean distance measure in RGB space, there is. In general, it can be said that D_E in RGB is sensitive to variations in intensity, and relatively much less sensitive to variations in hue and saturation [12].

3.3.2 Behavior in Other Color Spaces

The Euclidean distance characterizes color differences in each color space differently. Table 2 summarizes these results. It is clear that color differentiation in the RGB and XYZ spaces based on the Euclidean distance does not fully reflect the underlying necessity of including hue in the calculations. For example, the two colors, which are closest in hue, are furthest apart in Euclidean distance in the XYZ space.

From the above experiment, it can be said that the Euclidean distance seems to quantify perceptual differences in the CIELUV space similarly to the CIELAB space. This has also been observed earlier [50]. This result seem to strengthen the initial hypothesis and say that the combination of CIELAB with the Euclidean distance could account for both intensity and hue differences.

Color representations based on the dichromatic reflectance model such as rgb, $h_1h_2h_3$, and $l_1l_2l_3$ seem to characterize color differences well since the intensity component has been factored out

in some way. However, it seems odd that the Euclidean distance quantifies color differences in the $h_1h_2h_3$ space well although the dichromatic reflectance model does not support this conclusion. That is, Shafer's model states that the $h_1h_2h_3$ space should be invariant to highlights, and not illumination intensity. The seemingly good color discrimination based on hue of rgb (i.e., normalized RGB) and $l_1l_2l_3$ spaces for the example image in Figure 6 is directly in line with the results based on Shafer's model.

Color Space	**Euclidean Distance**			**Color Discrimination**
	D_E(A,B)	**D_E(C,D)**	**D_E(B,C)**	
RGB	59.72	59.69	60.34	Poor
XYZ	60	60	33	Poor
CIELAB	6.8	6.8	23.4	Good
CIELUV	7	7	37	Good
Rgb	0.0014	0.0024	0.12	Good
$l_1l_2l_3$	0.0054	0	0.77	Good
$h_1h_2h_3$	17.1	0	117.3	Good

Table 2: Comparison of Euclidean distance measures for different color spaces

3.4 Vector Angle

An alternate distance measure is the vector angle (D_{VA}) [12] defined as

$$\cos\theta = \frac{\vec{v}_1^{\,T}\vec{v}_2}{\|\vec{v}_1\| \cdot \|\vec{v}_2\|} \tag{18}$$

Unlike the Euclidean distance, the vector angle measure is insensitive to intensity differences, but quantifies well hue and saturation differences. However, there are certain drawbacks to using the angle θ as a similarity distance measure, including the problematic computation of statistics on values in angular coordinates [64].

A problem with using $\cos\theta$ or $1 - \cos\theta$ is that the dynamic range of values for small angles is small compared to the dynamic range for small angles when using $\sin\theta$. This is important since it is desirable to emphasize hue differences however small they may be. Therefore, the $\sin\theta$ was proposed in [12] as the actual angular distance measure and is defined as

$$D_{VA} = \sin\theta = \left(1 - \left(\frac{\vec{v}_1^T \vec{v}_2}{\|\vec{v}_1\| \cdot \|\vec{v}_2\|} \right)^2 \right)^{1/2} \tag{19}$$

If computational requirements were strict, one could use the $\sin^2\theta$.

It is assumed that all the color values are positive. This assumption will not hold for one of the color spaces examined in this thesis: $h_1h_2h_3$. However, experiments have shown that this is not of major concern. The computation of angles greater than 90° in spaces such as $h_1h_2h_3$ or in a translated RGB space will be examined in the future.

Only the sine of the vector angle will be considered in this thesis and will be referred to as D_{VA} or vector angle henceforth. The vector angle has also been independently presented for color image retrieval in its angular form [1].

The computational expense of calculating the $\cos\theta$ is 11 M/D's and 6 A/S's, as well as as well as 11 M/D's and 6 A/S's for each of two square roots (assuming a three term Taylor expansion) for a total of 33 M/D's and 18 A/S's. For the $\sin\theta$, this increases to 12 M/D's and 7 A/S's together with as well as 11 M/D's and 6 A/S's for each of three square roots for a total of 45 M/D's and 25 A/S's. However, due to simplifications the actual number of operations is 22 M/D's and 13 A/S's.

The properties of the vector angle measure will be discussed for the different color spaces with respect to the dichromatic reflection model. In general, it can be said that the vector angle measure is unstable near the origin of any color coordinate system.

3.4.1 RGB

Consider two pixel values $\vec{v}_1$ and $\vec{v}_2$ from areas with the same reflectivity characteristics in the RGB space, but under different illumination intensities (assuming equal color balance in the illumination). They are related as indicated by Equation 16.

Even though the two pixels come from areas with the same intrinsic color, the vector angle in RGB space would be exactly zero (irrespective of the variation in illumination); i.e.,

$$D_{VA} = \sin\theta = \left(1 - \left(\frac{\vec{v}_1^T \cdot \alpha\vec{v}_1}{\|\vec{v}_1\| \cdot \|\alpha\vec{v}_1\|} \right)^2 \right)^{1/2} = \left(1 - \left(\frac{\alpha\vec{v}_1^T \vec{v}_1}{\alpha\|\vec{v}_1\|^2} \right)^2 \right)^{1/2} = (1-1)^{1/2} = 0 \tag{20}$$

This shows a very desirable characteristic of this measure; namely, that the angle between two colors in the RGB space will be insensitive to variations in illumination (intensity), but sensitive

to differences in hue and saturation. The same result would be obtained by using the body reflectance term from Shafer's model. In other words, the invariance is only valid for matte surfaces. The vector angle should still be sensitive to glossy effects in the image (i.e., highlights).

This can be observed by examining the angle between the different colors from Figure 6. The angles between colors *A* and *B* and between color *C* and *D* are both 0.11°, which means that both sets of RGB values have approximately the same hues. However, the angle between colors *B* and *C* is 11.6°. A major problem with using the angle or sine of the angle as a distance measure between two colors is that the angle is undefined at the (0,0,0) point. Because of this, the calculation of vector angle when one of the vector magnitudes (or intensities) is small exhibits near random behavior. This issue will be examined again in Section 0.

3.4.2 Behavior in Other Color Spaces

The vector angle characterizes color differences in each color space differently from the Euclidean distance. Table 3 gives a summary of computation of vector angles and sine of the vector angles.

It is interesting to note that the vector angle measure gives good color difference information for all color spaces to be used with it. This is suggested by Shafer's dichromatic reflection model; however, there does not seem to have been any studies to evaluate whether the vector angle reflects perceptual differences better than the Euclidean distance. This should be part of future research. Several other conclusions can be reached from the example presented in Section 3.2.

The same problems found with using this measure on low intensity RGB colors would be encountered in the XYZ space since it is a linear transformation of the RGB space. The XYZ space might not benefit from the intensity invariant properties of RGB since the Y component represents the luminance of an object and Y is a weighted sum (cf. see Equation 4) of the RGB components.

The results from using the vector angle on the example image strengthen the hypothesis that CIELAB captures well color differences. It is not possible to use Shafer's model to extract meaningful information about the interaction of the CIELAB space and the vector angle. Therefore, nothing can be said about the effect of the color comparison done with the vector angle in CIELAB at this time. It is possible that because the computation will take into account the luminance component that the results will be worse than in RGB. Furthermore, it is not expected that the vector angle would add any capability to the CIELUV space given the presence just as in the previous case of a luminance component. Intensity invariance has been factored into the U and V components; however, because they are used together with the luminance, again it is not possible to say what positive effect the vector angle might have.

The results for the normalized RGB and the RGB spaces are identical. This is expected since the sum of the RGB components used in the normalization process to obtain the rgb space is factored out in the vector angle calculation. In practice, round-off errors might slightly change the differences being calculated.

The vector angle discriminates very well between colors of differing hues since invariance to highlights has already been taken into account in the design of the $h_1h_2h_3$ color space. The vector angle in this case might be a much better discriminator than the Euclidean distance since the color similarities between vectors *A* and *B* were not as well computed with the Euclidean distance. The vector angle further adds intensity invariance properties based on Shafer's model (i.e., just like in RGB).

However, the possibility of the vector angle being too sensitive to color differences for practical applications arises. For example, the distance between color points B and C is very high for both the $h_1h_2h_3$ and $l_1l_2l_3$ spaces which means that large amounts of noise could be introduced when processing images in those color spaces.

Color Space	**Vector Angle**						**Color Discrimination**
	D_{VA}(A,B)		**D_{VA}(C,D)**		**D_{VA}(B,C)**		
	θ	**sinθ**	**θ**	**sinθ**	**θ**	**sinθ**	
RGB	0.11°	0.002	0.11°	0.002	11.6°	0.20	Good
XYZ	0.20°	0.035	0.05°	0.000	6.3°	0.11	Good
CIELAB	0.64°	0.011	0.12°	0.002	14.8°	0.26	Good
CIELUV	0.95°	0.017	0.15°	0.003	22.9°	0.39	Good
rgb	0.11°	0.002	0.11°	0.002	11.6°	0.20	Good
$l_1l_2l_3$	0.44°	0.008	0°	0	66°	0.91	Good/Noisy
$h_1h_2h_3$	0.46°	0.008	0°	0	71.1°	0.95	Good/Noisy

Table 3: Comparison of vector angle measures and properties for color spaces

3.5 Distance Measure Combinations

An emerging area in color image processing is the combination of hue and intensity based information in color similarity measures. There are two principal ways of combining distance measures: using an addition-based approach [66] or a multiplication-based method [1]. Both present advantages and disadvantages; however, the main problem lies in deciding whether a color difference of 0.1 obtained with one distance measure and 0.1 obtained with another distance measure give an indication of similar color differences. Of course, this is not clear

however the answer to this question is beyond the scope of this thesis and should be part of future research.

In this thesis, both distance measures are normalized to fit within the zero-to-one range, which means that *a priori* the distances are compared on an equal basis. In the case of Euclidean distance, the calculations need to be normalized whereas the values for vector angle are already scaled within the 0-1 range since the sine of the vector angle is used. Only the addition-based combination technique will be examined in this thesis. For the multiplication-based combination, please refer to [1] as a start.

As the name implies, addition-based combination techniques focus on adding weighted values of the different distance calculations. They take the following basic form:

$$C = a_1 D_1 + a_2 D_2 + \ldots + a_n D_n \tag{21}$$

where C is the combined distance, D_i (i is an integer and $n \geq i \geq 1$) represents the i^{th} distance measure and a_i its relative weight. D_i can be either the result of applying a distance measure such as the Euclidean distance to different color spaces or the result of applying different similarity measures such as the Euclidean distance and the vector angle to the same color space (e.g. RGB).

It is necessary now to focus on obtaining the appropriate weights a_i for each distance calculation. In this thesis, only two distance measures will be considered; therefore, only two weights are needed. The weights can be arranged to be the opposites of each other; that is, if one of the weights is high then the other one should be low to always have a normalized output. Therefore, only one trade-off parameter would need to be calculated. The changing nature of the trade-off parameter will decide adaptively what determines the final relative weight of each distance measure in each color distance calculation.

Given that in this thesis the determination of a color difference is dependent only on the two colors being compared (i.e., the neighborhood of the pixels is not taken into account), two values have to be calculated (one for each of two points being compared) to obtain the trade-off parameter or distance measure weight. Therefore, a transition function between the different feature spaces is necessary for each point in the pair.

Let the value needed by the trade-off function be X_i for i=1,2. The sigmoid is a widely used transition function. A sigmoid function can be defined by

$$\alpha(X) = \frac{1}{1 + e^{-slope(X_i - offset)}} \tag{22}$$

where the *slope* defines the slope at the transition point and *offset* the transition midpoint. This function is shown in Figure 8. Both the *slope* and the *offset* are application-dependent and are

set experimentally in this thesis. In this case, as the parameter X_i increases, the function's output slowly changes from zero towards one biasing the outcome towards one of the calculations based on the underlying image characteristics.

However, since every time two points are considered, the trade-off parameter function [7] shown in Figure 9 is defined as

$$\rho(X_1, X_2) = \sqrt{\alpha(X_1) \cdot \alpha(X_2)} \tag{23}$$

This trade-off parameter has been first used by Carron and Lambert [7] and then by Wesolkowski and Jernigan [66].

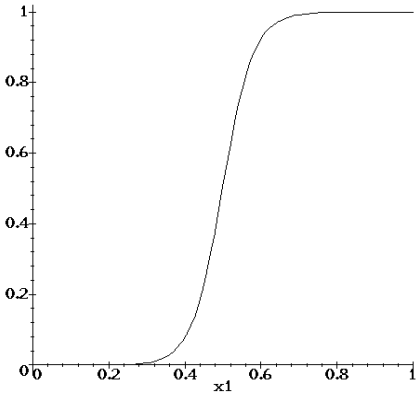


Figure 8: Trade-off parameter function

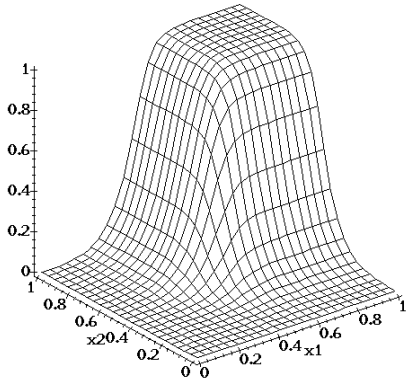


Figure 9: Example of equation (23)

There are several advantages of trying to combine distance measures. The shortcomings and advantages of the Euclidean distance, and the vector angle have been shown in previous sections. It would be beneficial if each of these measures could be applied when it is the most discriminating. Of course in the end, this depends very much on the application. That is, what are the distance measures needed by some application? For what is the application needed? If these questions can be successfully answered the choice of distance measures and potentially one of the combinations of distance measures would be quite easily chosen. This means that not all distance measures are necessarily going to perform well on all problems. This is an important observation, which will be discussed further in Chapter 6:.

In this thesis, two distance measures will be used: the Euclidean distance (an intensity-based measure in RGB), and the vector angle (a hue and saturation-based measure in RGB). If these color similarity measures are used, then the combined function with the trade-off parameter can be calculated knowing that

$$a_1 = \rho(X_1, X_2) \qquad (24)$$
$$a_2 = 1 - \rho(X_1, X_2)$$

where a_1 is the weight for the vector angle and a_2 is the weight for the normalized Euclidean distance. Then, the combined color similarity measure can be written as:

$$C = a_1 D_E + a_2 D_{VA}$$

or

$$C = \rho(X_1, X_2)\sqrt{1 - \left(\frac{\vec{v}_1^{\,T}(x,y) \cdot \vec{v}_2(x,y)}{\left\|\vec{v}_1(x,y)\right\| \left\|\vec{v}_2(x,y)\right\|} \right)^2} + (1 - \rho(X_1, X_2)) \cdot \left\|\vec{v}_1(x,y) - \vec{v}_2(x,y)\right\| \qquad (25)$$

In this case, the vector angle measure will be used only when X_i is high for both points, otherwise the Euclidean distance measure or a combination of both should be used. As the parameter X increases, the function's output slowly changes from zero (i.e. Euclidean distance bias) towards one (i.e. vector angle bias). The question of what X_i's should be now arises.

The possibility of using the intensity or the saturation parameters will be explored in the next two sections.

3.5.1 Intensity-Based Combination

A simple method of combining both distance measures would involve using the intensity of both points being compared. One way of calculating intensity involves taking a simple average of the RGB components. The use of intensity as a trade-off variable is a logical choice given that the vector angle measure breaks down for low values of intensity (see Section 3.3.1). Therefore, vector angle could be used when both pixels being compared have high intensity. Euclidean distance would be used when one of the two pixels would have low intensity.

To compute the intensity only one multiplication and two additions are required.

3.5.2 Saturation-Based Combination

A saturation-based combination of hue and intensity planes for edge detection was first attempted in [7]. Carron and Lambert converted the RGB color image into an HSI representation using the YC_1C_2 transformation as an intermediary step:

$$\begin{bmatrix} Y \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1 & -1/2 & -1/2 \\ 0 & -\sqrt{3}/2 & \sqrt{3}/2 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (26)$$

This results in the saturation definition shown in Equation 27.

$$S = \sqrt{C_1^2 + C_2^2} \tag{27}$$

They argued that this form of saturation is less sensitive to nonlinear effects than the classical saturation transformation [7]. Recall that this transformation [19] is given by

$$S = 1 - \frac{3 \cdot Min(R, G, B)}{R + G + B} \tag{28}$$

Carron and Lambert show that the noise variance within the hue component is higher than within the intensity component when saturation is low (i.e. intensity is more relevant than hue). They also show the noise variance within the intensity component is higher than within the hue component when saturation is high (i.e. hue is more relevant than intensity). In this thesis, Carron and Lambert's method will be used with saturation (as defined by Equation 27) and intensity as independent variables in the trade-off function.

The vector angle provides a good measure of hue difference and the Euclidean distance a good measure of intensity difference directly within the RGB space. This is potentially an improvement over using the hue and intensity planes from a complex transformation as shown in [7]. Therefore, when both pixels are highly saturated, vector angle would be used, and when one of the pixels is low in saturation, the Euclidean distance would be used.

To compute the intensity five multiplications and two additions are required together with 11 M/D's and 6 A/S's (assuming a three term Taylor series expansion) for the square root operator for a total of 16 M/D's and 8 A/S's.

3.6 Computational Complexity

Combining distance measures inherently increases the computation complexity of the color similarity measure since at least two different calculations have to be made. Furthermore, based on the nature of the independent variable (i.e., either intensity or saturation), the computational complexity increases further.

The computation requirements for each similarity measure calculation are given in Table 4. The computational complexity for the combination cases includes the computation of the trade-off parameter (i.e., the weights for both similarity measures) and assumes that a look-up table is available for the exponential function. The vector angle is about three times as computationally expensive as the Euclidean distance especially because of the normalization step. It is also easily noticed that the combined similarity measures are very computationally expensive in relation to either the Euclidean distance or the vector angle.

The computational complexity of the combination methods might not make them practical at this point. However, the ability to combine both intensity and hue information might warrant the use of such a computationally expensive algorithm to obtain increased performance.

Distance Measures	Independent Variable	Computational Complexity	
		M/D's	A/S's
Euclidean Distance	N/A	14	11
Vector Angle	N/A	22	13
Euclidean distance and vector angle (RGB only)	Intensity	55	39
	Saturation	69	43

Table 4: Comparison of computational complexity for color similarity measures

3.7 Summary of Distance Measures

In summary, the vector angle was shown to be a good hue-based discriminator of RGB color vectors; however, it is not very good at taking into account intensity. It actually does not consider intensity information at all. The Euclidean distance was found an inadequate color difference calculator in the RGB space. This measure takes into account all color information (i.e., intensity, saturation and hue) from RGB pixels to come up with the color difference whereas the vector angle only factors in the saturation and hue information.

For other color spaces, the interaction between the similarity measure and the properties of the color space is more complex. In color systems such as CIELUV and CIELAB, the measures seem to be equivalent color discriminators. It is seemingly easy to assess these properties from a theoretical perspective based on the dichromatic Reflectance Model. However, in practice the results might be quite different because certain assumptions (such as Equation 2) used in deriving Equation 3 might not hold at all illumination intensities. That is, the nonlinear relationship between the RGB color channels is not fully taken into account by some of these assumptions. So, the results presented in this chapter indicate a possible outcome for practical applications; however, the theory might not be adequate to provide good answers in this respect and the hypotheses formulated here might prove to be false. New assumptions might be needed to develop solutions where the current assumptions might break down.

From a computational perspective, it is clear that the Euclidean distance is by far the fastest way to compute color similarity; however, it might not be the best way.

In the next two chapters, two image processing applications will be examined: color edge detection and color image segmentation. The similarity measures examined here will be adapted to the algorithms presented therein.

Chapter 4: Color Image Edge Detection

4.1 Introduction

What is an edge? In gray-level images, edges have been typically modeled as brightness discontinuities. From an intuitive sense, it can be said that an edge is an apparent boundary between two pixels with significantly different brightness values. Here "significantly different" may depend on local pixel brightness statistics for example. This variation usually occurs because an edge usually represents a physical boundary between two objects having different intensities. The word edge is used to refer to a location on the image where the brightness value appears to jump. These jumps are associated with high values of the first derivative and are the kinds of edges that were originally detected by Roberts [47,20].

4.2 Gray-Level Based Techniques

Several edge detectors were developed in the 1960's and 1970's [19,20]: Roberts [47], Prewitt [45], and Sobel [53]. These simple yet powerful methods are still used today in many different applications although many other edge detectors have been developed [22]. Several of the standard gray-level edge detectors have been adapted to color image processing for this thesis. They will be briefly described here with their color extensions being described in the following sections.

The Roberts edge detector [47] is a simple approximation of the image differentiation operator. It is simply defined as

$$D_R = \sqrt{\big(v(x,y) - v(x+1,y+1)\big)^2 + \big(v(x+1,y) - v(x,y+1)\big)^2} \tag{29}$$

This implementation can also be described by two masks (see Figure 10). Equation 29 shows the computation of the Roberts operator using a cross difference. It is also possible to compute it using differences in the horizontal and vertical directions instead of the cross differences shown above.

1	0
0	-1

0	1
-1	0

Figure 10: Roberts Operator 2x2 Masks

The Prewitt edge detector [45] is an extension of the Roberts edge detector to a 3-by-3 neighborhood (see Figure 11). Here, the horizontal and vertical differences are computed.

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Figure 11: Prewitt Operator 3x3 Masks

The Sobel operator is the classical edge detector standard [53]. It is very similar to the Prewitt edge detector. It is described in Figure 12. The Sobel operator emphasizes the horizontal and vertical differences between the pixels closest to the central pixel.

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Figure 12: Sobel Operator 3x3 Masks

A more robust edge detection algorithm is commonly referred to as the Canny edge detection method [6]. The Canny operator is a sophisticated gradient-based edge detection algorithm. It will be further explained in Section 4.4.3.

4.3 Color-Based Edge Detection: Literature Review

For color images, the notion of an edge is much more complex than in grayscale images. In color images, intensity, hue and saturation of a color all play a part in determining object boundaries. A physical boundary produces an edge which needs to be captured using a measure that combines the different color characteristics. The concept of color similarity now becomes important since pixel intensities alone cannot be used to determine the existence of an edge (see Chapter 3: for more details).

The use of color in edge detection increases the amount of information needed for processing which complicates the definition of the problem. For grayscale images, most edge detectors use local gradient information or a difference operator in some fashion. In color images, a distance measure (typically the Euclidean distance) needs to be used to define the color gradient. It is, therefore, very easy to extend the Roberts', Prewitt's, Sobel's, and Canny's methods by replacing the difference operators by the Euclidean distance, vector angle or a combined measure.

The color image processing extension to the Roberts [12], Sobel and Canny [16] operators will be described in the next section. First, a literature review outlines the work recently done in color image edge detection.

For color images, a number of approaches have been proposed from processing individual planes [7,24] to true vector-based approaches [12,18,36,50,52,57,58,70,71]. The computational load of computing edges on individual planes can be much smaller than that of computing edges on the color vector. However, this seems to be a trade-off between speed and algorithm performance. The vector-based approaches exploit the correlation between the color planes much more effectively than the computation on single planes. This is why most researchers have concentrated on the vector-based approaches.

The Sobel operator has been applied successfully to all three planes in the RGB space and the gradients were summed to obtain the resultant edges in [24]. Hedley and Yan compute the Sobel operator on each of the three RGB planes and then sum the results. For their map processing application where colors and objects are well defined, this seems to be an adequate technique for edge detection. However, for more complex color images where it is necessary to capture better the correlation between the planes, this approach would probably be inadequate.

The Sobel operator was also applied to each component of the HSI space and the individual results were combined using a trade-off parameter between hue and intensity [7]. An interesting feature of this trade-off parameter was its dependence on the level saturation. The results of this combination are not convincing given the test images used (there are only minor differences between results where hue information is used as compared to those where it is not). Color image scenes containing shadows might have provided a better indication of the capabilities of combining the information contained within the individual HSI planes.

Several researchers have applied vector order statistics methods such as vector mean and vector median filters [70], the minimum vector dispersion (MVD) edge detector [58], and the vector range operator [57] in the RGB space. However, again it seems that the inappropriate image test sets were used to show the viability of the methods. The authors did not compare their methods to a vector-based Sobel or Canny operator. This would go a long way in demonstrating that their methods are truly superior at least when intensity is used as the prevalent source of information for color edge detection (i.e., when using the Euclidean distance similarity measure).

Another approach for edge detection was the calculation of the maximum Euclidean distance between the central pixel and all its neighbors, which is called the vector gradient [71]. This algorithm was found to work best in the CIELUV space [50]. However, this determination seems to have been done on an ad-hoc basis since no methodology for choosing the best result was provided in [50]. Further, no methodology for identifying a threshold to detect the actual

edge was given. However, this method is simple and elegant. It will be described in the next section in more detail.

Another method for finding edges in multidimensional images approaches the problem from an eigenvector point of view [71]. In this method, the eigenvectors for a 3x3 neighborhood are computed and the distance between them is calculated using the Euclidean distance. This value gives an indication of the strength of an edge at that point. This method was successfully further used by Gevers and Smeulders on several different color spaces [18].

The entropy operator [52] has also been used for color edge detection. It calculates the entropy in a local region based on RGB color vectors. The entropy is calculated individually for each of the three planes. The total entropy at the central pixel is defined to be the weighted sum of these individual entropy values. The weights are defined as the normalized RGB values (cf. see Section 2.3.5 for a definition of the normalized RGB space). One of its interesting properties is that it yields a small value when the color chromaticity in a region becomes uniform and produces a large value when the color chromaticity is highly changing.

Moghaddamzader et al. [36] describe a hybrid method which uses RGB and $Lh_{ab}c_{ab}$ (see Figure 3), a hue-based derivative of the LAB space which the authors refer to as the HSI space in their paper. The authors use fuzzy membership functions based on intensity and saturation information to assign a relative value to the hue contrast. Next, hue contrast is normalized using this information. Finally, the Euclidean distance between the RGB color vectors is averaged with the hue contrast measure producing an edge detector based both on hue and intensity differences. The method seems quite complicated given what it is trying to accomplish (i.e., color edge detection). In addition, the results presented are not convincing since they are shown on images that do not necessarily present a problem in the color domain.

4.4 Color-Based Edge Detection Algorithms

In this thesis, several edge detection algorithms adapted for color image processing will be compared. The algorithms are the modified Roberts operator [12], the Sobel operator, the Canny operator [6], the Vector Gradient operator [50], and the 3x3 Difference Vector operator [70].

The example image of Figure 6 is used here to illustrate the difference between an edge detection performed with a Euclidean distance-based edge detector and one based on the vector angle. Applying an Euclidean distance-based operator yields the result shown in Figure 13. It can be noticed that all apparent edges in the image have been detected. The threshold used was very low (i.e., a value of 5 out of a range of 0-255) to allow the displaying of all significant edges detected in the image. The operator distinguishes between all regions. However, when a vector angle-based edge detector is applied, the result is an image such as the one shown in

Figure 14. It is immediately apparent that the edges between regions A and B and regions C and D are non-existent. This is due to the vector angle being near zero between the values in these two color pairs. The value of this will be further illustrated on test images in Chapter 6.

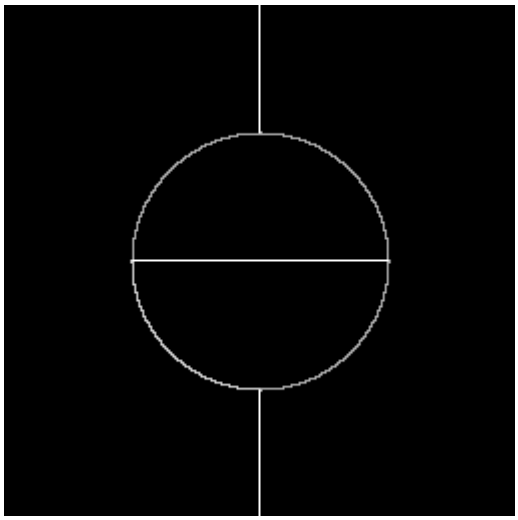


Figure 13: Edge detection using a Euclidean distance-based operator

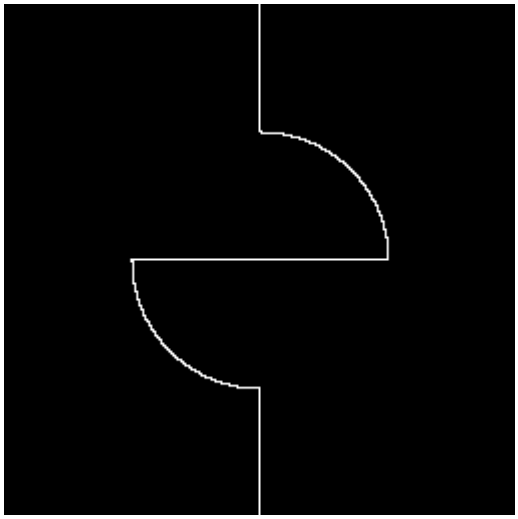


Figure 14: Edge detection using a Vector Angle-based operator

4.4.1 Modified Roberts Edge Detectors

A very simple edge operator is the Roberts operator [47]. A modified version of this operator [12] calculates the maximum absolute difference between diagonally adjacent pixels in a 2x2 block instead of computing the magnitude of the gradient (cf. Equation 29). This operator can be generalized to multidimensional pixel values by:

$$E_R = \max\left(D_E\left(\vec{v}(x,y), \vec{v}(x+1,y+1)\right), D_E\left(\vec{v}(x+1,y), \vec{v}(x,y+1)\right)\right) \quad (30)$$

where $\vec{v}(x,y)$ is the vector containing the multiple values of the pixel at coordinate (x,y).

The modified Roberts operator can also be extended to the proposed vector angle difference. The vector angle modified Roberts can be calculated using the maximum sine of the angles between diagonally adjacent pixels in a 2x2 block as

$$S_R = \max\left(D_{VA}(\vec{v}(x,y), \vec{v}(x+1,y+1)), D_{VA}(\vec{v}(x+1,y), \vec{v}(x,y+1))\right) \quad (31)$$

The use of the combined distance measure presented in Section 0 parallels that of the Euclidean distance and vector angle-based modified Roberts operators described above.

The Roberts edge detector is a simple operator. The vectorization of this operator allows for a better performance on color images than of the intensity values alone (i.e., summing edge detection values obtained independently from each of the different color planes). However, this edge detector lacks the robustness of the Sobel or other more sophisticated operators such as the Canny edge detector.

4.4.2 Sobel Edge Detector

As mentioned earlier, the Sobel operator is a very well known edge detector [53]. In this thesis, the original difference-based gradient computation as shown in Figure 12 is replaced by a Euclidean distance calculation. This vectorization of the algorithm allows for the effective use of the color information given that simple intensity differences would not represent differences between two color vectors as well as a Euclidean distance calculation.

The Sobel operator has been shown to be a good edge detector. In its expanded form, it will deal better with the information contained in color images without compromising it such as in methods where the operator is applied to each color plane independently [24]. In those cases, the correlation between the various planes is lost and the final result would be probably less than adequate. The Sobel operator will suffer from an inability to identify all hue difference-based edges just as other Euclidean distance-based operators.

The Sobel edge detection operator will be applied to the different color space.

4.4.3 Canny Edge Detector

The Canny edge detector [6] uses linear filtering with a Gaussian kernel to smooth the noise in the image. Next, the edge strength and direction are calculated for every pixel in the smoothed image. The Canny operator does this by differentiating the image in the horizontal and vertical directions, and then computes the gradient magnitude as the root sum of squares of the derivatives. The arctangent of the ratio of the derivatives is used to compute the gradient direction. The next step is called nonmaximal suppression. In this process, the edge strength of

each candidate edge pixel is set to zero if its edge strength is not larger than the edge strength of the two adjacent pixels in the gradient direction. The pixels that survive the nonmaximal suppression thinning process are labeled as candidate edge pixels. An adaptive thresholding method is then applied on the thinned edge magnitude image to obtain the final edge map.

Adapting this algorithm to color images is not trivial. One could apply the Canny operator to each plane of a color space. However, just as was mentioned in the previous cases this would result in a lower performance than when the color information from all planes is used at the same times. In this thesis, an adaptation to color images by Gauch using cubic splines to estimate the derivatives is applied [16]. The non-maximal suppression and edge map thresholding steps have been transformed into a zero-crossing detection step with an adaptive threshold relative to the local gradient.

In its original form, the Canny operator is known for emphasizing weak edges and yet suppressing edge output due to noise. As will it be shown in Chapter 6, the Canny operator can perform very well when one is trying to detect edges due to intensity changes. However, when edges occur due to hue differences, this might not be the case anymore. The Canny operator will show similar results to other Euclidean distance-based edge detectors.

The Canny edge detector will be only applied to gray and color RGB images.

4.4.4 Vector Gradient Edge Detectors

The vector gradient edge detector is a local operator which computes the maximum distance with the desired similarity measure between the center pixel and the 8-connected pixels adjacent to it [71]. It has already been used successfully with the Euclidean distance measure in the LUV space [50].

The Euclidean distance version of this operator can be simply defined as

$$E_{VG} = \max_{i=1\ldots8}\{\|\vec{v}_i(x,y) - \vec{v}_0(x,y)\|\} \tag{32}$$

where i is a counter representing each of the eight neighboring pixels (see Figure 15) and $\vec{v}_i(x,y)$ is the i^{th} color vector in the neighborhood of the examined color point at position (x,y).

The vector angle version of this edge detection operator is written as

$$S_{VG} = \max_{i=1\ldots8}\left[\sqrt{1-\left(\frac{\vec{v}_i^{\,T}(x,y)\cdot\vec{v}_0(x,y)}{\|\vec{v}_i(x,y)\|\,\|\vec{v}_0(x,y)\|}\right)^2}\right] \tag{33}$$

1	2	3
8	X	4
7	6	5

Figure 15: Positions of neighborhood pixels

The formula for the vector gradient operator with a combined distance measure would be

$$C_{VG} = \rho(X_1, X_2)S_{VG} + (1-\rho(X_1, X_2))E_{VG}$$

or

$$C_{VG} = \max_{i=1...8}\left(\begin{array}{l}\rho(S_1, S_2)\sqrt{1-\left(\dfrac{\vec{v}_i^{\,T}(x,y)\cdot\vec{v}_0(x,y)}{\|\vec{v}_i(x,y)\|\ \|\vec{v}_0(x,y)\|}\right)^2} \\ +(1-\rho(S_1, S_2))\cdot\|\vec{v}_i(x,y)-\vec{v}_0(x,y)\|\end{array}\right) \tag{34}$$

The vector gradient uses the central pixel as the reference in the gradient calculation and chooses the maximum difference as the edge gradient. This is not necessarily beneficial, as there can be a lot of noise in an image in contrast to the Roberts, Sobel or Canny operators which use some kind of implicit averaging in their calculations.

4.4.5 3x3 Difference Vector Edge Detectors

A well-known edge detector in image processing is the 3x3 difference vector (DV) edge detection operator [70], which is characterized as a 3x3 operator calculating the maximum gradient across the central pixel. The Euclidean distance version of the difference vector operator can be written as

$$E_{DV} = \max_{i=1...4}\{\|\vec{v}_i(x,y)-\vec{v}_{4+i}(x,y)\|\} \tag{35}$$

where i represents one of the first four (out of a possible eight) positions around the central pixel, whereas $4+i$ represents each of the positions opposite to the first four (see Figure 15). This is done in order to obtain measurements for the four directional gradients (i.e., horizontal, vertical, left and right diagonals) across the central pixel. $\vec{v}_i(x,y)$ is the i^{th} color vector in the neighborhood of the color point at position (x,y).

The vector angle version of the difference vector edge detector is characterized by

$$S_{DV} = \sqrt{1 - \max_{i=1...4}\left(\frac{\vec{v}_i^{\,T}(x,y) \cdot \vec{v}_{4+i}(x,y)}{\left\|\vec{v}_i(x,y)\right\| \left\|\vec{v}_{4+i}(x,y)\right\|} \right)^2} \tag{36}$$

The difference vector operator using the saturation-based combination measure would be represented by

$$C_{DV} = \rho(X_1, X_2) S_{DV} + (1 - \rho(X_1, X_2)) E_{DV}$$

or

$$C_{DV} = \max_{i=1...4}\left(\begin{array}{l} \rho(S_1, S_2) \sqrt{1 - \left(\frac{\vec{v}_i^{\,T}(x,y) \cdot \vec{v}_{4+i}(x,y)}{\left\|\vec{v}_i(x,y)\right\| \left\|\vec{v}_{4+i}(x,y)\right\|} \right)^2} \\ + (1 - \rho(S_1, S_2)) \cdot \left\|\vec{v}_i(x,y) - \vec{v}_{4+i}(x,y)\right\| \end{array} \right) \tag{37}$$

Notice that the maximum is being computed on the whole gradient calculation. This is done in order to conserve the relative meaning of the pixels. That is, the Euclidean distance Difference Vector operator calculation is to be combined with its vector angle Difference Vector operator counterpart.

The DV operator, just like the gradient vector operator, does not do any implicit or explicit averaging of the differences. Therefore, one can expect that there will be also some noise since again only the maximum value is selected as the gradient. However, it will be twice as fast as the vector gradient operator due to the reduced number of distance calculations (i.e., one half the number of distance calculations).

Suggestions for using larger neighborhoods for the difference vector operator are discussed in [70] while vector angle adaptations are suggested in Chapter 7.

4.5 Computational Complexity

This section details the computational complexity of using a particular distance measure within an edge detection technique in a particular color space. Combined measures will also be treated here since their computational load is quite a bit higher than that when only single distance measures are applied.

An important consideration in a practical system is the computational complexity of an operation. The lower the computational complexity, the faster the algorithm will be executed. In real time systems, differences of milliseconds can be crucial. The complexity of using the various distance measures together with particular color spaces adds another factor in choosing which distance measure and color space to use for edge detection in a practical system.

The computational complexity for each pixel assuming three components per color vector is summarized in Table 5. It is immediately noticed that the computational load of combining similarity measures is very large compared to non-combining approaches. The computational complexity also increases as the number of Euclidean distance and especially vector angle calculations is increased. Thus, the individual vector gradient edge detectors will take longer to execute than their counterparts in other edge detector groups.

Finally, the additional computation incurred by smoothing the image for the purpose of edge detection is not included in this calculation except for the gray level version of the Canny algorithm since it is part of the algorithm definition.

Edge Detection Method	**Distance Measure**	**Computational Complexity**			
		ED's	**VA's**	**M/D's**	**A/S's**
Modified Roberts	ED	2	0	28	22
	VA	0	2	44	26
	Combination/Intensity	2	2	110	78
	Combination/Saturation	2	2	138	86
Sobel	Single plane (graylevel)	N/A	N/A	15	13
	ED	6	0	97	73
Canny	Single plane [6]	N/A	N/A	31	40
	ED [16]	6	0	69	63
3x3 Vector Gradient	ED	8	0	112	88
	VA	0	8	176	104
	Combination/Intensity	8	8	440	312
	Combination/Saturation	8	8	552	344
3x3 Difference Vector	ED	4	0	56	44
	VA	0	4	88	52
	Combination/Intensity	4	4	220	156
	Combination/Saturation	4	4	276	172

Table 5: Computational compelxity of edge detection methods

4.6 Edge Thresholding

When detecting edges, it is important to find an appropriate edge intensity threshold beyond which significant (i.e., meaningful for the segmentation of objects within an image) edges are present. Without a threshold or with a very low threshold, a lot of noise edges might appear that

would make it more difficult to perform further image understanding tasks such as object recognition.

In this thesis, a global threshold is used. The best threshold is determined based on the quality of the edge detection as determined by Pratt's Figure of Merit [44] (please see Chapter 6 for an explanation of this quality measure). This cannot be done when applying these algorithms to real world problems. However, in this thesis the objective is to compare the best possible performances of the different edge detectors. It is assumed that Pratt's Figure of Merit will indicate which threshold gives the result closest to the ideal edge map.

The only exception to this will be the threshold used for the Canny operator. The thresholds used in the Canny operator are locally adaptive and therefore try to take into account the variations within each region. The thresholds used for the grayscale and color image Canny operators were determined experimentally for the particular images used.

Finally, the design of an adaptive thresholding algorithm is left as a future research topic and is discussed in Chapter 7.

4.7 Summary of Edge Detection

The edge detectors that will be used in this thesis have been summarized. They include the modified Roberts operator, the Sobel operator, the Canny algorithm, the vector gradient operator and the 3x3 difference vector operator. Based on the computational analysis, the vector gradient method will take the most time and the modified Roberts operator the least amount of time to execute. The vector gradient or other color edge detection operators would have to have a better performance to justify using them over the modified Roberts operator.

Algorithms such as the Sobel operator and the modified Roberts operator have been originally adapted for color processing in this thesis. The algorithms based on the vector angle have been shown to perform as expected on an artificial test image. These edge detectors will be applied to real scene images to test their robustness and performance quality in Chapter 6.

In the next chapter, color image segmentation will be addressed by studying a new algorithm for color clustering.

Chapter 5: Color Image Segmentation

5.1 Introduction

Image segmentation is probably the most important task in image understanding. It is the partitioning of an image into a set of non-overlapping regions whose union is the entire image. The purpose of image segmentation is to decompose the image into parts that are meaningful with respect to a particular application. Without good image segmentation, it is not possible to process the image appropriately and, therefore, to understand what it represents.

It is very difficult to define what constitutes a "meaningful" segmentation of an image within a computer algorithm. Haralick and Shapiro suggest that the following rules are usually obeyed [20]:

1. Segmented regions should be uniform and homogeneous with respect to some characteristic such as gray level or texture.
2. Region interiors should be simple and without many small holes.
3. Adjacent segmented regions should have significantly different values with respect to the characteristic on which they are considered uniform.
4. Boundaries of each segment should be simple, not ragged, and must be spatially accurate.

However, achieving these desired properties is very difficult since applying strictly uniform and homogeneous measures to segmentation produces regions full of small holes (due to intensity variation or compression artifacts in images [73]) which have ragged boundaries. Furthermore, trying to create regions with large differences in values of the measured properties can cause highly similar regions to merge and boundaries to be lost.

Image segmentation is the process of grouping data into different spatial aggregates. Haralick and Shapiro suggest that there is no full theory of clustering, and, therefore, no full theory of image segmentation [20]. This means that image segmentation techniques are generally ad hoc and differ on how they emphasize one or more of the desired properties. In the end, each method tries to balance one property against another. Therefore, the final implementation of each image segmentation algorithm depends very much on the end application. However, these differences usually center on the choices of parameters or methods of how to adapt certain parameters to the image.

In this thesis, one variant of image segmentation algorithms will be explored: color clustering. Clustering does not segment pixels based on their spatial relationship, but rather based on their

intensity or other inherent characteristic. Many algorithms enhance clustering algorithms with spatial constraints to make them more practical; however, in this thesis, only the general clustering algorithms will be discussed. Another type of image segmentation known as region growing will not be explored. Region growing is a method which aggregates neighboring pixels around a seed pixel and keeps growing the region as long as certain homogeneity criteria are satisfied.

The purpose of discussing image segmentation in this thesis is to illustrate how various distance measures in color images can help in improving the segmentation process. The premise is to show differences in color image segmentation based on different criteria which in the end would be application-specific.

The color image segmentation field can be roughly divided into three categories: global or color clustering methods [3,40,56,67,68,69,70], local or region growing algorithms [8,15,59,72], and hybrid procedures [3,24,29,32,49,73]. These algorithms base their principles on physical models of the image [29,56,67] or on the statistical nature of images [5,9,34,50]. This thesis will concentrate on studying the impact of using different color clustering similarity measures within the color clustering context.

5.2 Histogram-Based Grayscale Image Clustering

Clustering in pattern recognition is the process of partitioning a set of pattern vectors or data into subsets called clusters. There are many clustering algorithms detailed in the literature such as k-means, ISODATA, etc. [13,48]. The most difficult aspect of clustering is the cluster validation problem. That is, it is difficult to either estimate the number of clusters the data is made up of or to determine the particular characteristics of every distinct data partition. The first problem is usually solved in an ad hoc manner by overestimating the number of clusters and then using cluster merging criteria. The second problem can be solved by selecting thresholds based on neighborhood or global statistics with respect to cluster size, variance of the data, etc.

The most referred to and simplest clustering algorithm is the k-means algorithm [48]. This algorithm iteratively computes the mean of a set number of clusters until it converges to a stable set of cluster prototypes (i.e., means). This algorithm is described in detail in Section 5.4 below.

In gray-scale images, regions are typically modeled as uniform intensity areas and segmentation algorithms employ some form of Euclidean distance measure to determine pixel similarity either on a spatially local basis or on a global color basis. The clustering problem reduces to a thresholding or multilevel thresholding problem. That is, for example, in binarization the question becomes what threshold will partition the image into two well-separated background and foreground clusters [37]. Such partition might allow the further stages in a system to

identify parts of the image as text and apply optical character recognition algorithms to them. For more complex images, a multilevel thresholding method could be used to discern several intensity clusters in the data [38]. This could be used in partitioning grayscale images into many different regions only based on their intensity or some intensity-related property.

5.3 Histogram-Based Color Image Clustering

For color image processing, the clustering algorithms operate in complex multidimensional spaces. Because of the added complexity of needing three variables to represent color pixels, the issue of region segmentation in color images is not as well defined as for gray-scale images. The segmentation now focuses more on creating regions of homogeneous color. This means that the choice of distance measure becomes very important since homogeneity depends very much on how distances between colors are being measured. In this case, all approaches found in the literature use some form of Euclidean distance to determine similarity between two color pixels.

A number of color clustering approaches have been proposed. Most approaches have used the RGB space for applying their algorithms [24,32,70,73]. Other approaches are based primarily on one color space: CIELAB [68], and CIELUV [49]. Some approaches were also designed for non-standard hybrid multidimensional spaces [69]. A few algorithms were designed and tested with a variety of color spaces [34,40]. A method for color image segmentation evaluation based on the Euclidean distance measure has also been devised [3].

Next, several global image segmentation techniques will be reviewed. In addition, some hybrid methods implementing a global approach as a first step will also be described.

Yang describes a method based on the classification of a vector composed of two angles and the magnitude describing the color pixel RGB triplet [70]. The method uses two thresholds to determine region homogeneity: comparing the Euclidean distance between the angles of two vectors to a "vector direction tolerance" and comparing the magnitude to a second threshold. For two vectors to be considered similar in color, both the Euclidean distance between the angles representing the directions of each vector and the vector magnitudes have to be within a tolerance threshold of each other. This does allow the use of the hue component for region segmentation purposes; however, the use of the Euclidean distance measure is awkward and thresholds are needed. Only one image was used in testing this method; therefore, it is not known how accurate it is.

Park et al. describe an approach based on morphological transforms [40]. They first smooth the 3-D RGB histogram using a difference of Gaussians and then apply closing and dilation operators to segment the image. The clusters follow a non-uniform expansion (i.e., a cluster could be greatly extended in the horizontal direction but not in the vertical direction). Again,

the cluster formation process is decided based on the Euclidean distance thereby not taking into account hue very well. Their results on an artificial image with shadows show promise. However, it is questionable whether their results on the standard house image show that their algorithm is superior to the k-means algorithm for example.

Wu et al. use a fuzzy c-means clustering algorithm for preliminary segmentation of RGB map images for the purpose of extracting lines and text [69]. A neural network is then trained on a set of cluster centers, which satisfy a set of criteria. The image is finally segmented using the neural network optimized cluster centers using a nearest neighbor approach. Again, the use of the Euclidean distance as the similarity measure is problematic.

Several other methods make use of a global segmentation approach as a preliminary step to more specific region segmentation or merging. For example, Hedley and Yan [24], as well as Zhou et al. [73], use distance measures which incorporate a spatial component to complement their color clustering approaches in order to mitigate the effects of outliers and anti-aliasing. This is a very important feature of the algorithm, however, the authors have only applied their algorithms to images where the intensity is constant. These algorithms would fail if applied to images with shadows since they use the Euclidean distance or its variants for region similarity computation.

Schettini applies a histogram-directed clustering approach based on a recursive one-dimensional histogram analysis to identify spatial regions of uniform color [49]. The technique oversegments the image and then uses a merging technique to collapse spatially neighboring zones. This means that the segmentation is done on a statistical basis at best and not a physical understanding of the image formation process. This is especially important if the regions are of constant, but different hues, and, therefore, belong to different objects. It would be necessary to implement complex region merging algorithms to remedy this problem.

Kurugöllü and Sankur show a method, which exploits the correlation present in 2-D histograms of RGB pairs (i.e. RG, GB and RB) to perform region clustering [32]. The region segmentation is very dependent on the intensity. This can be clearly observed on the only result image provided (the color “Lena” image). It is concluded that this type of segmentation cannot deal well with intensity changes.

Furthermore, Woelker implements a color clustering method in CIELAB space with each cluster being defined as a cube with a Euclidean distance-based tolerance [68]. This Euclidean distance tolerance needs to be small if dissimilar colors of similar intensity are not to be included in the same region.

In this thesis, the k-means approach will be examined in detail together with the Mixture of Principal Components (MPC) neural network [67]. The effect of clustering data based on a

single pixel, as well as a 3x3-pixel neighborhood will be examined. Since this section is only concerned with the comparison of using either the Euclidean distance measure or the vector angle measure, the results of the application-specific approaches in the literature will not be examined at this time.

5.4 Euclidean Distance-Based Clustering: The K-Means Algorithm

The k-means (KM) algorithm is defined as follows [48]:

1. Choose the number of classes, *k*.
2. Choose initial cluster centers $\vec{\mu}_1, \vec{\mu}_2, \ldots, \vec{\mu}_k$.
3. Classify each data point.
4. Recompute estimates for $\vec{\mu}_i$ using the results of 3.
5. If $\hat{\mu}_i$ are consistent, stop; else go to step 3.

Step 3 is carried out using a simple 1-nearest neighbor approach [48] using the Euclidean distance as a similarity measure. Typically, the straight distance between two points is the most intuitive similarity measure. The number of classes, *k*, and choosing initial cluster centers, $\vec{\mu}_1, \vec{\mu}_2, \ldots, \vec{\mu}_k$, are still two challenging problems. In this thesis, the number of classes for k-means to operate on is defined and the initial cluster centers are chosen randomly with an appropriate distribution for the color space being used in the clustering process.

The Euclidean distance is very suitable for the k-means algorithm since this distance measure can be directly related to the computation of the cluster centers or means. For example, if we take two points and average them, the distances from each of these points to the mean will be equal. If there are more points in the average, then the distance from each of those points to the mean will be proportional to the contribution of that point towards the average of the data; i.e., the vector mean.

Since the use of the Euclidean distance as a similarity measure assumes that the color space is isotropic, the clusters defined by this distance will be invariant to translations or rotations [13]. However, they will be sensitive to linear and nonlinear transformations in general, especially those that distort the distance relationships. Given that the RGB space for example is not perceptually uniform, the assumption that it is isotropic fails and, therefore, clustering algorithms using the Euclidean distance will not necessarily yield good results.

The k-means algorithm was implemented using the standard batch-based approach. That is, the means were recalculated after all training patterns had been classified. There exists an online

training algorithm for k-means [46]; however, given their equivalence, it was decided that the batch-based training approach would be implemented. Finally, to avoid problems with the class means falling into local minima, the k-means algorithm was trained with a progressively increasing number of classes (e.g., first 2 classes, then 4, etc. until the desired number of classes was reached). This ensured that the class prototypes actually corresponded to means representative of the data.

5.5 Vector Angle-Based Clustering: The MPC Approach

The choice of the similarity measure biases in a subtle manner the results of a clustering algorithm [13]. For example, the Euclidean distance and its variants have long been used for image segmentation whether global or local in nature. However, the Euclidean distance between two colors in RGB does not necessarily reflect the visual separation between these two colors [12,50] (see discussion in Chapter 3). Therefore, a distance measure which would quantify hue and saturation, might be more suitable for processing color images. The vector angle measure has been recently introduced for that purpose [12] and was detailed above in Section 3.4. The vector angle has been suggested as a similarity measure in the past for clustering algorithms [13].

For color clustering, the vector angle cannot be substituted directly for the Euclidean distance in the k-means algorithm. Recently, a mixture of principal components (MPC) has been developed to represent data in a new way [10,11]. This approach has been applied already to color image segmentation [67].

The difficulty with directly substituting the vector angle for the Euclidean distance lies in the way k-means calculates the class means or prototypes. For k-means, the class prototype is the average of all the vectors in the class. Distance calculation with Euclidean distance measure is directly related to the mean vector. From a intuitive point of view, the contribution of a color point to the vector mean is directly proportional to the distance between those points. The analogous notion of a prototype for the vector angle is the principal eigenvector of the class. The principal eigenvector (i.e., the eigenvector corresponding to the largest eigenvalue) indicates the most prominent direction of the data cluster in the given space. Intuitively, the principal eigenvector appears to be the angular analogy of the vector mean, just as the vector angle is analogous to the Euclidean distance. The MPC effectively combines the vector angle and principal component extraction into a clustering algorithm.

Figure 16 illustrates the modular architecture of an MPC network for a single basis vector per class. Each module consists of a basis vector and represents a class of the input data. Therefore, if a single color vector is used for clustering within each module, it will be made up usually of three values (i.e., one corresponding to each of the elements in the color vector).

However, for multispectral images, there will be a corresponding increase in size of the color basis vector.

A basis vector corresponds to the principal eigenvector for a given class thereby defining the class. Each basis vector performs a linear transformation on the input data. The input vector, $\vec{x} \in \Re^N$, is linearly transformed by each of the *K* modules resulting in *K* coefficient vectors, $\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_K$. A one-dimensional linear subspace is defined by each single basis vector, $\vec{w}_i$, contained in the corresponding module. So, if $\vec{w}_i$ is of dimension *N*, the coefficient y_i is calculated as

$$y_i = \vec{w}_i^T \vec{x} \tag{38}$$

In the MPC approach, the raw dot product is used rather than the actual vector angle form to compute the distance between the prototype cluster centers and the data. This is because the normalization of the result is done implicitly by the classifier. That is, $\|\vec{w}_i\| = 1$ and $\|\vec{x}\|$ is constant across all *i*.

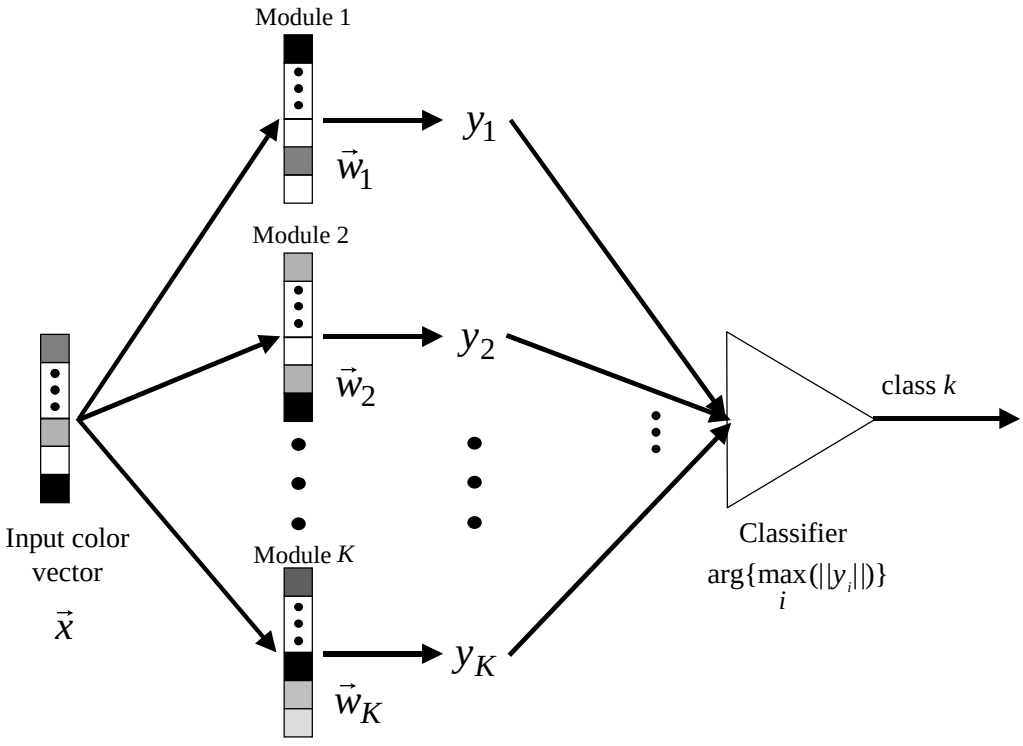


Figure 16: Coding section for MPC network with K classes and 1 component per subspace

The classifier then chooses the output of the winning module based on the subspace classifier

$$\arg\{\max_i(||y_i||)\} \tag{39}$$

For a network with unit norm basis vectors, this classifier is equivalent to maximizing equation 3, i.e., minimizing the vector angle between the basis vector, $\vec{w}_i$, and the input, $\vec{x}$. In other words, the principal eigenvector of the class closest to the color point being classified using the vector angle measure will be chosen.

In summary, the class is represented by the basis vector which defines the prototypical angle. Contrast this with k-means, which uses the mean as the class prototype. In other words, the calculation of the class prototype using the vector mean for the k-means algorithm is analogous to computing the first principal component for the MPC approach [67]. Similarly, the calculation of color similarity using the Euclidean distance for the k-means algorithm would be analogous to computing this similarity using the vector angle for the MPC method. Therefore, the use of the vector mean within the k-means clustering algorithm to calculate the prototype of the class values while using the vector angle as a distance measure is inappropriate. Instead, the principal angle of the class is required, or equivalently, the first principal component of the class data. Finally, the clusters would be invariant to rotations and dilations (i.e., scaling) and not invariant to translation and general linear transformations [13].

To compute the principal components while refining the class definitions, an iterative algorithm is employed:

1. Choose the number of classes, *K*.
2. Initialize the *K* principal components, $\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_K$, to some appropriate set of values.
3. Classify a data point using the subspace classifier (minimum vector angle) using equations 38 and 39
4. Modify the basis vector, $\vec{w}_i$, of the winning class using a Hebbian neural network algorithm which extracts the class principal component [23,25,39].
5. If no convergence, go to 3, else stop

Just as in the k-means algorithm, determining the number of clusters and their position is still an open problem. In this thesis, the number of classes for the MPC approach is defined and the initial cluster centers are chosen randomly with an appropriate distribution for each of the different color spaces.

The MPC approach was implemented in an online or sequential training fashion. That is, the class prototypes were updated after each training pattern had been classified. This was

necessary since the batch-based approach proved to be highly unstable which prevented the algorithm from properly converging. It was therefore not very practical.

The sequential algorithm used was based on Oja's rule for unsupervised learning [39]. Oja's learning rule is simply a procedure for Hebbian learning with constrained weight vector growth. It adds a weight decay that is proportional to the squared output. Oja's learning method finds a unit weight vector maximizing the mean square output. This is equivalent to finding the principal component for zero mean data. Since in this thesis the color spaces are considered to have their origin fixed at (0,0,0), the learning algorithm finds the principal eigenvector of the autocorrelation matrix for each class with respect to the origin [10,11,39]. This is of course not necessarily useful for some spaces. However, the determination of which spaces need to be translated to obtain better classification results is left as a future exercise.

5.6 Summary of Color Image Segmentation

Two color clustering algorithms have been described in detail: the k-means algorithm and the Mixture of Principal Components method. The former is based on the Euclidean distance while the latter is dependent on the vector angle.

Since the vector angle measure used in the MPC algorithm is implicitly normalized the runtime computation per point and per prototype is reduced to the dot product between two vectors; i.e., 6 multiplication and 3 additions. This is much less than the computation of the Euclidean distance between the two vectors. Therefore, the MPC algorithm will be faster than the k-means algorithm in test mode (i.e., after all class prototypes have been computed).

However, in training the situation is slightly different. Given that the convergence properties of both algorithms were not studied, this analysis is beyond the scope of this thesis. The convergence speed of several subspace-based learning methods is presented in [27].

Chapter 6: Results and Evaluation

6.1 Experimental Set-Up

To test the various color similarity measures with the myriad of edge detection, and color clustering algorithms, a database of eight images has been collected. All images represent a scene from the real world. They are shown in Figure 17.

Image 1 was staged to contain shadows, highlights and inter-reflections. It is a composition of various toys and house items. The scene is made up of numerous vivid colors. Its well defined edges will help in the quantitative comparison of various edge detectors as well as the quantitative evaluation of the MPC/VA and the KM/ED algorithms for image segmentation. Image 2 shows an image of a person with colorful clothes on seashore background. Image 3 shows an outdoor scene with some objects such as a fire hydrant made up of well defined colors. Image 4 shows a complex scene of a group of people. Image 5 shows a flower on a purple wall background. Image 6 shows an image of a red metallic Volkswagen. Image 7 is the standard "peppers" image while Image 8 is the standard "Lena" image.

This chapter will first discuss results for the edge detection problem. Next, results from the color clustering approaches will be summarized. Since this thesis is only concerned with the comparison between the functioning of particular algorithms with the Euclidean distance and the vector angle, no extended effort has been made to produce a full system using either of these distance measures. This is left as an exercise for future work.

6.2 Edge Detection

6.2.1 Introduction

Several color edge detection operators were tested with the Euclidean distance: the modified Roberts operator, the Sobel operator, the Canny operator, the 3x3 vector gradient operator, and the 3x3 difference vector operator. Some color edge detectors were tested with the vector angle as the color similarity measure: the modified Roberts operator, the 3x3 vector gradient operator, and the 3x3 difference vector operator. All edge detectors are applied to all the different color spaces except for the Canny operator and the various operators based on combined distances which are studied only within the context of the RGB color space. The Sobel and Canny operators will be applied to the gray-level images for comparison purposes.

The first experiment consisted in using the color pixel triplet for classification. The edge detectors were applied to a smoothed image. The image was smoothed with a 3x3 averaging operator after which a transformation from RGB to another color space was used.

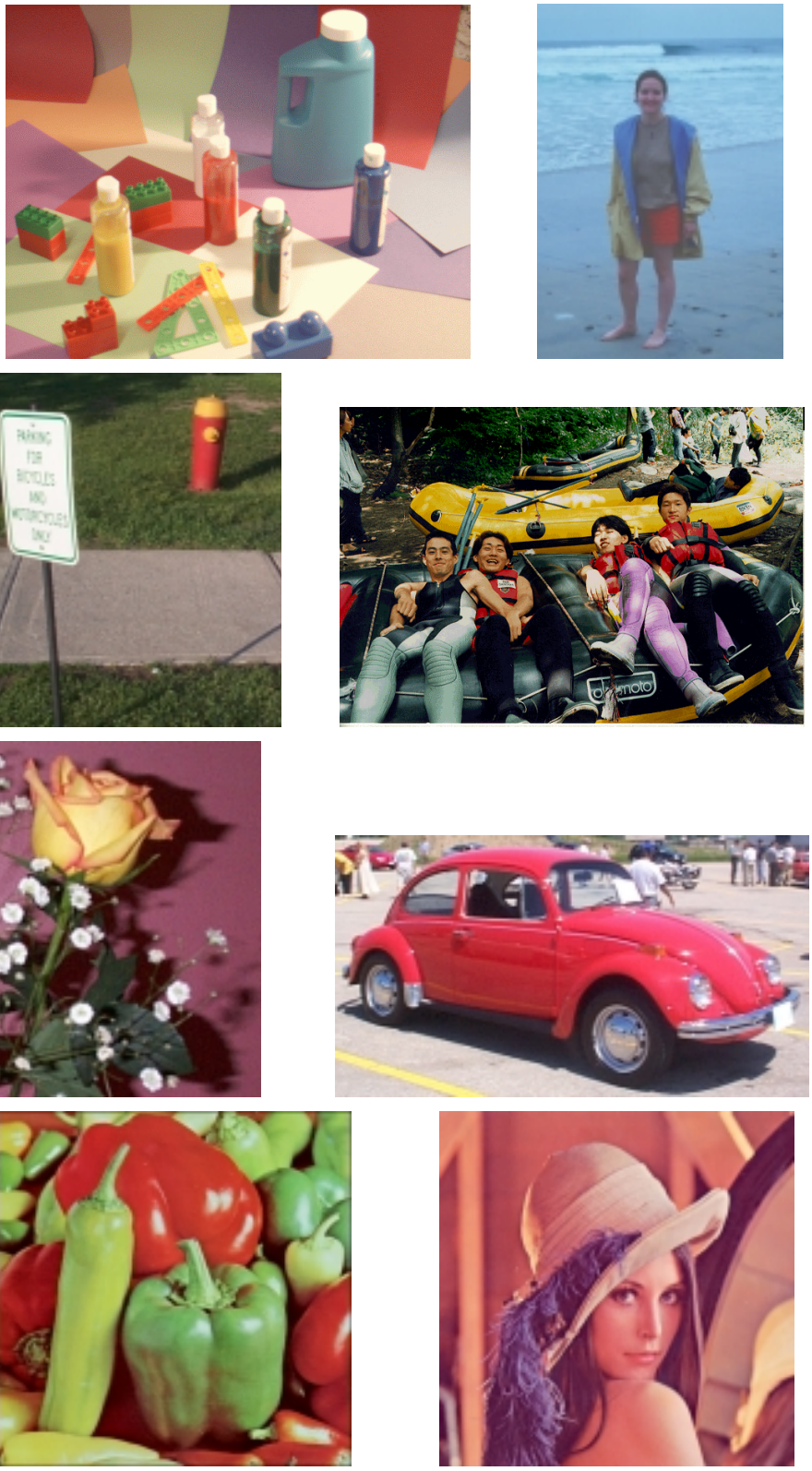


Figure 17: Images 1-8 used for testing

The experiment consisted in using the color pixel triplet for classification. The edge detectors were applied to a smoothed image. The image was smoothed with a 3x3 averaging operator after which a transformation from RGB to another color space was used.

The next two sections will detail the quantitative and qualitative results for edge detection. The quantitative results are based on Pratt's Figure of Merit [44]. The qualitative results are based on a visual comparison of the edge images obtained for the largest Figure of Merit value for that image.

6.2.2 Quantitative Evaluation: Pratt's Figure of Merit

Several quantifiable comparison methodologies exist for edge detection [21,22,44]. Heath et al. propose an edge detection comparison technique based on visual assessments by a group of people trained to assess edge quality [21,22]. In one study, the Canny operator is compared with the Sobel operator, the Nalawa-Binford edge detector, as well as the Sankar-Boyer edge detection operator [22]. This comparison method is based on the premise that human rating experiments can be done in a much more rigorous manner to provide useful quantitative conclusions. This paradigm is based on experimental psychology and statistics and assesses whether there is a statistically significant difference in edge detector outputs as perceived by humans when considering an object recognition task. Although this technique would be appropriate for evaluating color edge detectors, it was not used due to its complexity and inability to easily factor out extraneous edges. In fact, it is necessary to train people to detect exactly what one wants to be detected. However, problems could occur if people did not exactly understand how each edge contributes to the overall quality of the image. Instead, the well-known Pratt's Figure of Merit [44] was used to compare edge detector output. This measure is well understood and it is possible to control largely what is being tested.

Pratt's Figure of Merit (FOM) attempts to balance three types of errors that can produce erroneous edge maps: missing valid edge points, failure to localize edge points and classification of noise fluctuations as edge points. The Figure of Merit is defined as [44]

$$R = \frac{1}{I_N} \sum_{i=1}^{I_A} \frac{1}{1 + ad^2} \tag{40}$$

In this equation, I_N is the maximum of I_A and I_I. I_A represents the total number of actual edge pixels; i.e., those edge pixels that were found. I_I represents the total number of ideal pixels in the image; i.e., the number of edge pixels in the reference image (e.g. Figure 18 see for an ideal edge map of Image 1). The parameter a is a scaling constant while d is the distance from an actual edge point to the nearest ideal edge point (in this thesis a = 0.9). The scaling factor is used to penalize edges that are localized, but offset from the true position (in this thesis a scaling value of 0.9 is used). The rating factor is normalized so that a value of one means that the edge

has been detected perfectly. The Figure of Merit is normalized with the maximum of the actual and ideal number of edge pixels in order to ensure a penalty for smeared (i.e., $I_I < I_A$) or fragmented edges (i.e., $I_I > I_A$).

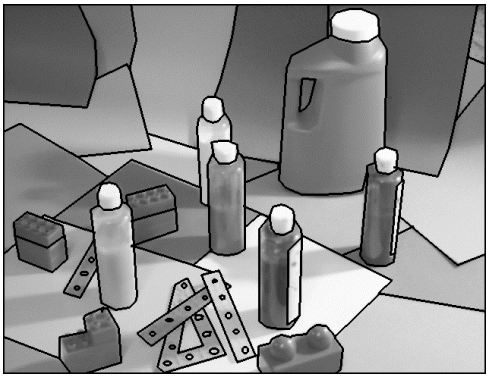


Figure 18: Hand-segmented truth image.

There are several problems associated with using Pratt's Figure of Merit. For example, the scene image might be incorrectly segmented to provide an accurate representation of an ideal edge. However, although the hand-segmentation can have some inaccuracies, the same edge map is compared across all image results. This ensures that all result images are compared in an equivalent fashion. It will become apparent to the reader that the Figure of Merit is a good indicator of edge detection performance. This will be demonstrated in the section below. Furthermore, an advantage to using Pratt's Figure of Merit is that extraneous edges can now be considered as noise since they are not be included in the ideal edge map.

In this thesis, Pratt's Figure of Merit was used to determine an ideal edge threshold. This was done by choosing the threshold value corresponding to the highest Figure of Merit score for each algorithm/color space combination. In this way, all results are compared to each other when they are closest (from a Figure of Merit point of view) to the ideal edge map. This ensures that the results being compared are relatively the best possible for that edge detection algorithm and color space combination.

The typical training run to obtain an edge map was as follows:

1. Convert the RGB image into a different color space if required
2. Perform edge detection operation on color image
3. Normalize edge values to the 0-255 range
4. Apply Pratt's Figure of Merit at all threshold levels (i.e., from 0 to 255)
5. Choose the image corresponding to the highest Figure of Merit value as an illustration of a good edge detection with the edge detection operator being tested

6.2.3 Staged Scene Image: Qualitative and Quantitative Results

The quantitative edge detection evaluation was done on Image 1 (see Figure 17). The Figure of Merit plots are shown in Figure 19 and Figure 20 for the modified Roberts operator. A list of thresholds with corresponding FOM values is shown in Table 6 for edge detection results from all operators on Image 1. These numbers give an indication of how much the edge image is close to an ideal edge map. The Figure of Merit tries to balance the effect of erroneous edges with the number of good edges. If there are many error edges and good edges then the edge image will be deemed of poor quality due to the high amount of noise. They will be discussed in the text as needed.

The Figure of Merit results indicate that the best edge detection results for the Euclidean Roberts are obtained with the $h_1h_2h_3$, RGB and XYZ color spaces. For the vector angle case, the best edge detection results are obtained with the RGB (and rgb), CIELAB and XYZ color spaces according to the Figure of Merit results. The rgb and RGB curves are superimposed because the normalizing factor in the denominator of the rgb values is eliminated due to the division in the vector angle calculation.

The thresholded edge maps for RGB and XYZ are shown in Figure 21. The Euclidean distance results are shown on the right while the vector angle results are placed in the left column (consequent figures are arranged in a similar manner). Given the quantitative result for the vector angle case explained above, it is not unexpected that the edge map obtained using the rgb space is identical to its RGB counterpart. The results for CIELAB and CIELUV are shown in Figure 22 while results for rgb, $h_1h_2h_3$ and $l_1l_2l_3$ are shown in Figure 23.

It is clear from examining the results that the edge detection methods using either the Euclidean distance or the vector angle are not clearly superior to one another. For example, in the RGB case the vector angle Roberts operator shows better discrimination in areas of differing hues while the Euclidean distance Roberts operator is better at finding edges between areas of different intensities. Since there are more such areas in the image, the Figure of Merit value is higher for the Euclidean Roberts operator. However, it could be argued that the vector angle result looks better in general since there are no shadow edges present in the image. The results in the XYZ space reflect closely the results in the RGB space.

The results for the CIELAB are relatively good with the vector angle result showing most of the edges. The edges that are missing are the same as those in the RGB space (e.g. white bottle in the center and colored paper in right bottom corner). However, the “blue bottle” is well formed in the vector angle case. The results do not seem as good as the RGB results, which agrees with the Figure of Merit result. The CIELUV results are disappointing (especially the edges around the “blue bottle”). These results are reflected in the FOM quantitative assessment as well.

Algorithm	Color Space	Max Figure of Merit	Algorithm	Color Space	Max Figure of Merit
Euclidean Distance Modified Roberts Operator	RGB	0.678746	Vector Angle Modified Roberts Operator	RGB	0.647929
	XYZ	0.657035		XYZ	0.602809
	CIELAB	0.618543		CIELAB	0.629472
	CIELUV	0.636672		CIELUV	0.566058
	rgb	0.632845		rgb	0.647929
	$h_1h_2h_3$	0.679676		$h_1h_2h_3$	0.531151
	$l_1l_2l_3$	0.531151		$l_1l_2l_3$	0.531356
Modified Roberts Operator	Combined / Intensity off=.88, s=2	0.693960	Modified Roberts Operator	Combined / Saturation off=.85, s=2	0.689219
Euclidean Distance Vector Gradient Operator	RGB	0.684174	Vector Angle Vector Gradient Operator	RGB	0.663736
	XYZ	0.659277		XYZ	0.627299
	CIELAB	0.641172		CIELAB	0.642540
	CIELUV	0.640500		CIELUV	0.584877
	rgb	0.647755		rgb	0.663736
	$h_1h_2h_3$	0.692398		$h_1h_2h_3$	0.532147
	$l_1l_2l_3$	0.532147		$l_1l_2l_3$	0.532374
Vector Gradient Operator	Combined / Intensity off=.50, s=1	0.698636	Vector Gradient Operator	Combined / Saturation off=.10, s=1	0.694955
Euclidean Distance Difference Vector Operator	RGB	0.664119	Vector Angle Difference Vector Operator	RGB	0.672654
	XYZ	0.647398		XYZ	0.628742
	CIELAB	0.616945		CIELAB	0.652867
	CIELUV	0.623495		CIELUV	0.595476
	rgb	0.626205		rgb	0.672654
	$h_1h_2h_3$	0.672252		$h_1h_2h_3$	0.547970
	$l_1l_2l_3$	0.512463		$l_1l_2l_3$	0.547789
Difference Vector Operator	Combined / Intensity off=.85, s=1	0.702619	Difference Vector Operator	Combined / Saturation off=.85, s=1	0.695673
Sobel	Gray	0.624518	Sobel	rgb	0.673525
	RGB	0.682067		$h_1h_2h_3$	0.709257
	XYZ	0.661203		$l_1l_2l_3$	0.574823
	CIELAB	0.665053		CIELUV	0.639175
Canny	Gray	0.344098	Canny	RGB	0.470092

Table 6: Quantitative Color Edge Detection Results

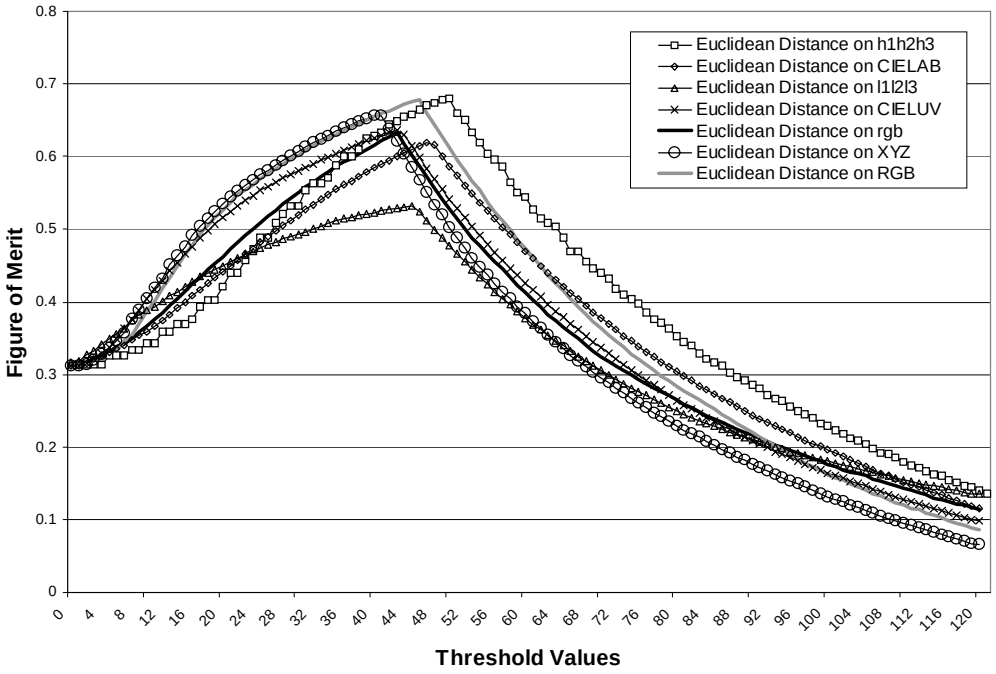


Figure 19: FOM values at thresholds 0-120 for Euclidean Roberts operator

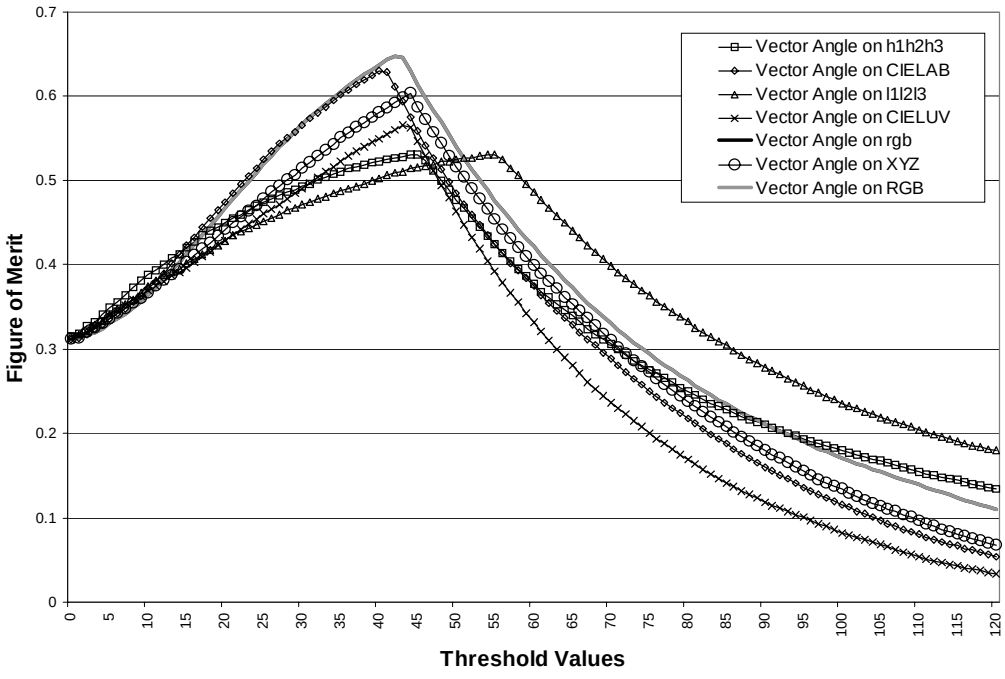


Figure 20: FOM values at thresholds 0-120 for Vector Angle Roberts operator

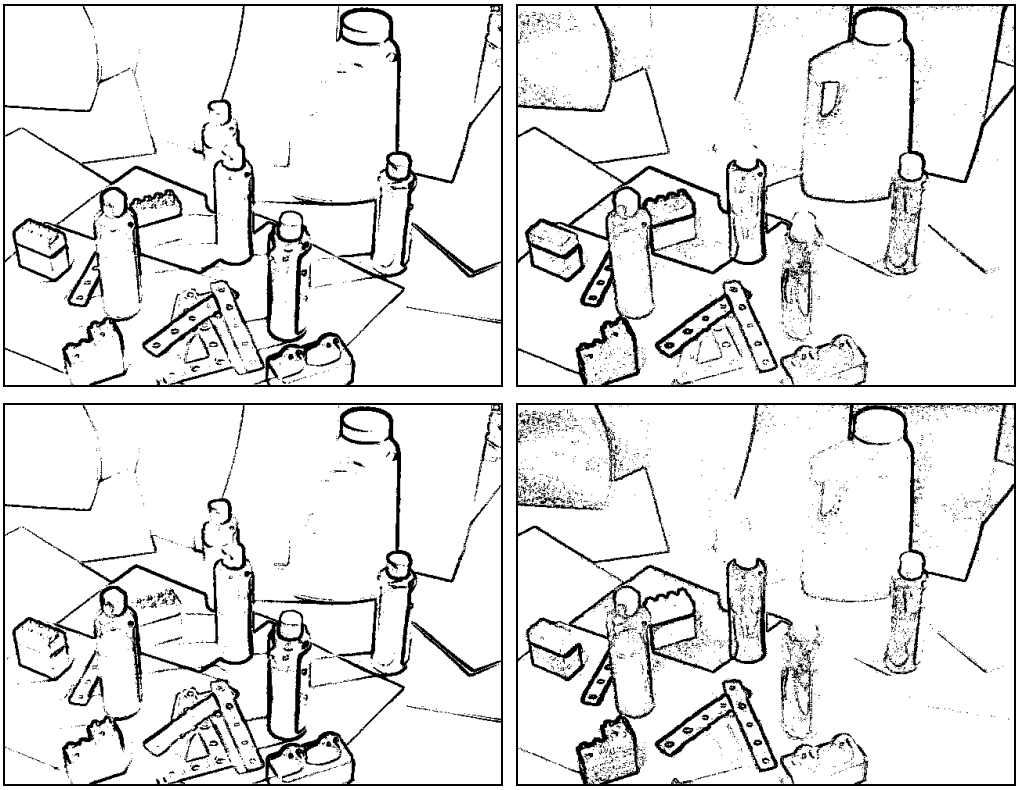


Figure 21: Img 1 RGB (top) and XYZ (bottom) results for the Roberts operator

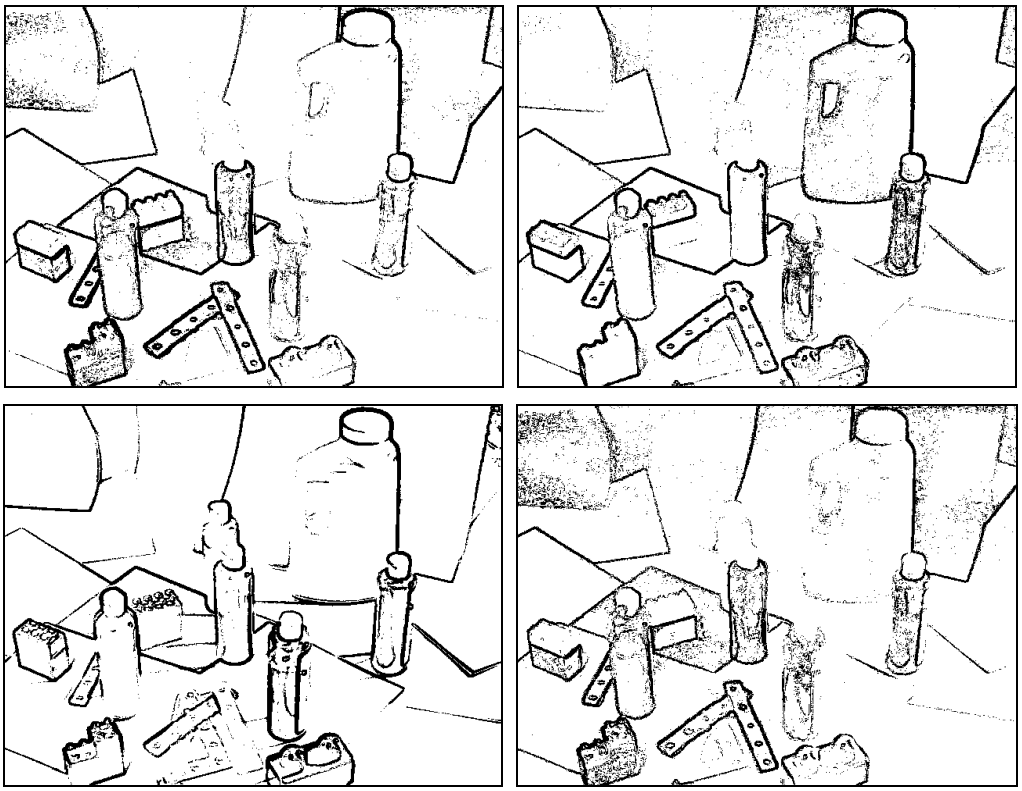


Figure 22: Img 1 CIELAB (top) and CIELUV (bottom) results for Roberts operator

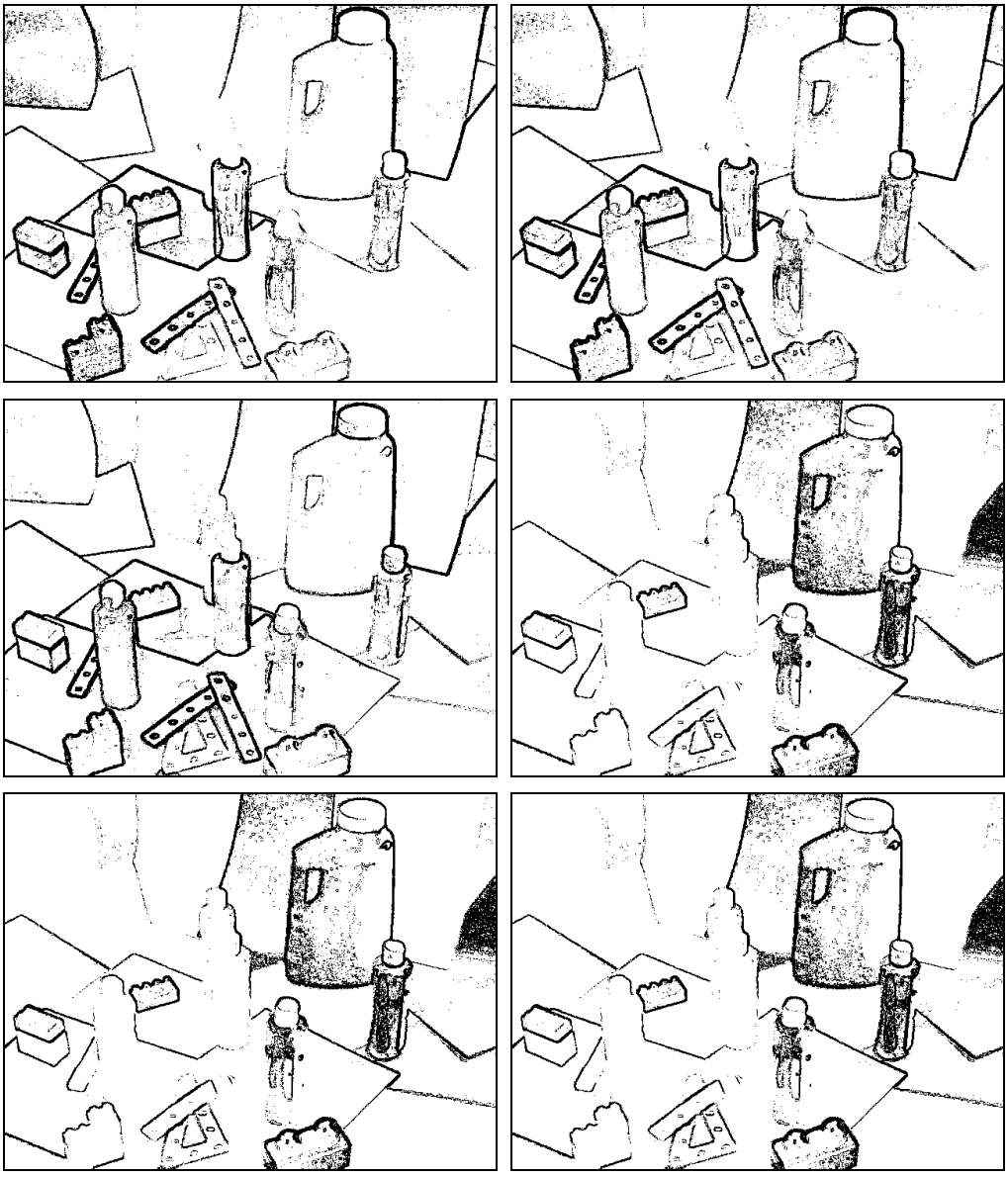


Figure 23: Img 1 rgb (top), $h_1h_2h_3$ (middle) and $l_1l_2l_3$ (bottom) results for Roberts operator

The best edge detection result without combining similarity measures visually (and quantitatively) is obtained with the $h_1h_2h_3$ color space with the Euclidean distance. The outline of all edges appears in the edge image while some edges are shown prominently. Excessive noise seems to affect the vector angle result (especially on the right side of the image). This is because of the compounding effect of having RGB values near the gray axis (which produces low values in $h_1h_2h_3$) and the unpredictability of the vector angle for vectors with magnitude close to zero. This is reflected in a below average Figure of Merit assessment. The results in the $l_1l_2l_3$ color space are also corrupted with noise which greatly affects the edge detection result. The poor visual edge detection performance is echoed in the Figure of Merit assessment.

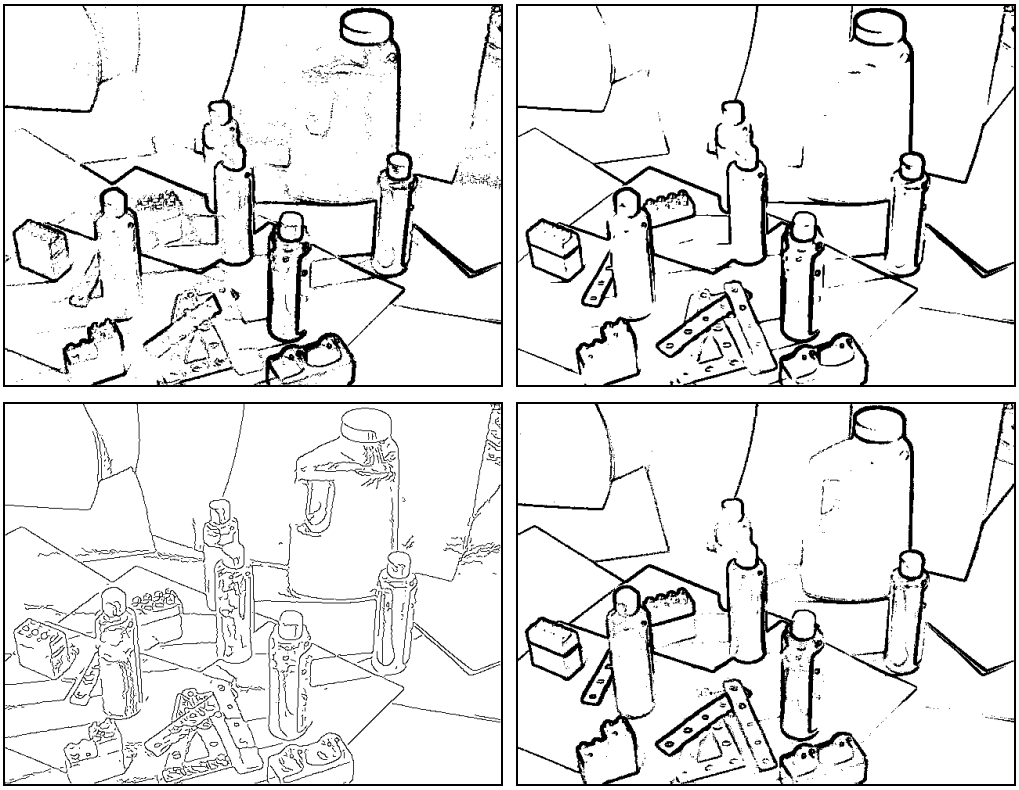


Figure 24: Image 1 resluts for Sobel (top) and Canny (bottom) operators (left - grayscale, right - RGB color)

The edge detection with the Sobel and Canny operators on a grayscale version of the image and the RGB color image is illustrated in Figure 24. Both operators in both modes show many extraneous edges which results in a low Figure of Merit score. However, the results on the RGB color images are much better than the results for the grayscale image which is expected since more data is available in a color image. The RGB Canny result is arguably better than the other results given that its FOM score is the highest of the four images and visually very few extraneous edges are shown. The edges for the blue bottle are faint but better than for Sobel and the other operators when the Euclidean distance is used.

Results obtained with the Sobel operator on other color spaces are shown in Figure 25. The best edge detection result according to the FOM score is obtained on the $h_1h_2h_3$ color space (a score of 0.709257). This is confirmed with a visual inspection with almost all edges being present in total or in part. This is a surprising result given that the $h_1h_2h_3$ space was one of the spaces with the least number of invariances derived from the Dichromatic Reflectance Model. The other edge detection results are not as good, with the $l_1l_2l_3$ color space yielding the worst edge detection.

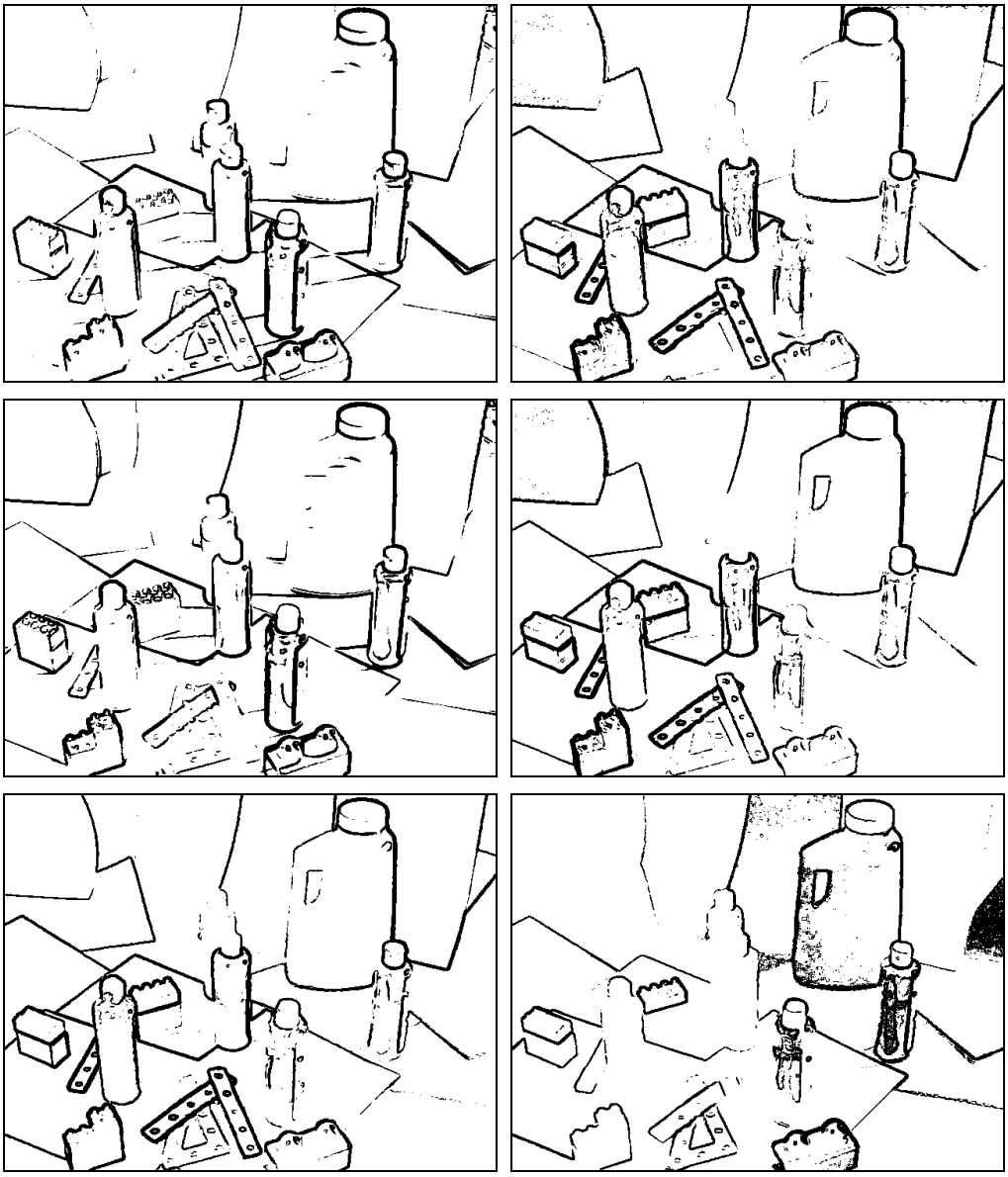
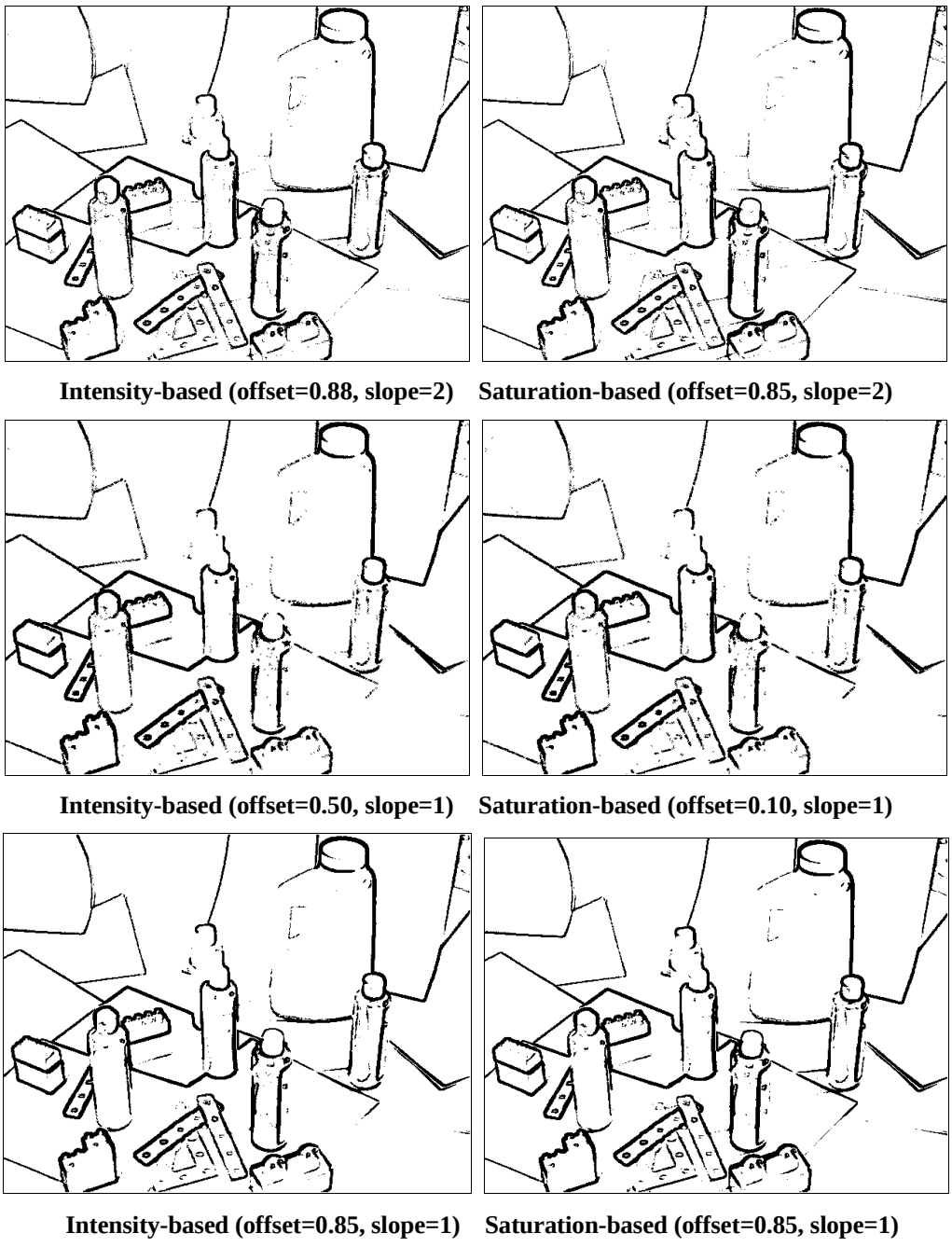


Figure 25: Image 1 resluts for Sobel operator (left to right): XYZ, CIELAB, CIELUV, rgb, $h_1h_2h_3$ and $l_1l_2l_3$

Finally, it is important to discuss the performance of edge detectors with a combined similarity measure. The quantitative results for the different edge detectors indicate that these are some of the best edge detection results within this set of algorithms. However, upon examination of the edge images (see Figure 26), the results appear only slightly better at best. In all results, part of the edge of the blue bottle is missing and most importantly, the edge formed by the piece of paper in the bottom right hand corner is not visible in some cases. Surprisingly, the modified Roberts operator gives better qualitative (but not quantitative!) results than the two more sophisticated algorithms (i.e., vector gradient and difference vector operators).



Intensity-based (offset=0.88, slope=2) **Saturation-based (offset=0.85, slope=2)**

Intensity-based (offset=0.50, slope=1) **Saturation-based (offset=0.10, slope=1)**

Intensity-based (offset=0.85, slope=1) **Saturation-based (offset=0.85, slope=1)**

Figure 26: Image 1 combined distance results in RGB for modified Roberts (top), vector gradient (middle) and difference vecotr (bottom) operators

It seems that the Figure of Merit might not be the most appropriate way to analyze optimal edge detection results which are to be viewed by humans since it does not directly correlate with the way humans perceive edges [21]. Humans tend to tolerate more noise. For example, Figure 27 shows an edge detection result with a lower threshold than the result shown in Figure 26. The image does not appear noisy and yet its FOM is much lower than before (the new threshold generates a FOM of 0.651872 versus the old 0.702619). However, given that the result is a

machine vision application which might not tolerate noise to the same extent that a human would, then perhaps the FOM gives a good indication of edge detection performance. Furthermore, the best way to assess the results of an edge detector is by studying its effect on the application for which it is meant. The Figure of Merit trades off showing more correct edges versus not showing erroneous edges thus leading to images that show mostly correctly detected edges and little erroneously detected edges.

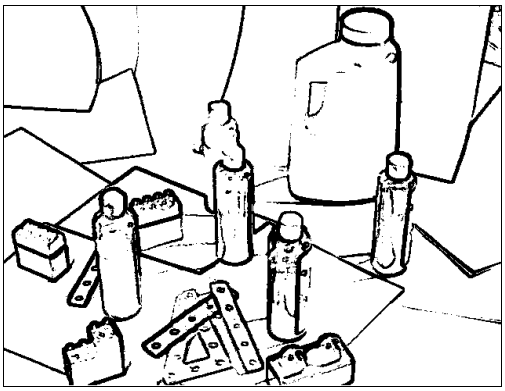


Figure 27: Difference Vector edge detection with the intensity-based combined measure (offset=0.85, slope=1) and a lower threshold than that given by FOM

6.2.4 Summary of Results

A surprising result is the one obtained using the vectorized Sobel operator and the modified Euclidean distance Roberts operator on the $h_1h_2h_3$ space. This result indicates that $h_1h_2h_3$ has to some degree desirable properties such as intensity-invariance.

The vector angle versions of the different edge detectors performed as expected by being more effective at separating areas where hue differences were more significant than intensity differences. However, due to the high noise propensity of this similarity measure many results have excessive amounts of noise.

The combination of Euclidean distance and vector angle measures helps to bridge the gap between intensity-based and hue-based differences in RGB. The results obtained using the combination methods enhance both measures to some degree; however, this increase would not be warranted given the very high computational load required. This is especially visible in areas of the image with a high intensity or saturation.

In the realm of combined distances, there are still numerous problems to be resolved. First, the problem of proper measure normalization arises. In this thesis, the distances are assumed to be equivalent in both measures. This is a preliminary step to discovering whether there is some kind of correspondence between the two. Does a 0.1 measurement within the normalized Euclidean distance measure mean the same as 0.1 within normalized vector angle measure?

Certainly not. This is a fundamental question which was not answered in this thesis. An answer to this question could well determine if this is a worthwhile endeavor.

Second, in this thesis, one combination method with two variants was used (intensity- and saturation-based combinations). This parallels research in [7]. A mix of saturation and intensity might achieve better combination results since intensity is needed to make sure hue is not used for low RGB pixel values and saturation is needed to decide whether two pixels are highly saturated and, therefore, more likely to have stable hue values.

Third, as mentioned before, the application is very important when assessing the results of an algorithm. In this case, edge detection constitutes only a preliminary step in an image understanding process. The edge detectors shown here should be evaluated in a broader context to verify that their functioning is consistent with our preliminary results. To this effect, more tests should be carried out on artificial and real images to fully assess the usefulness of the methods. In order to accomplish this effectively, it is also necessary to devise new quantitative evaluation methods that directly relate to the application that will use this preprocessing step.

6.3 Color Clustering

6.3.1 Introduction

Two different experiments were performed on different color spaces to evaluate the effectiveness of the new color clustering scheme. The color spaces used for the k-means algorithm were RGB, XYZ, CIELAB, CIELUV, rgb, $h_1h_2h_3$, and $l_1l_2l_3$. The color spaces used for evaluating the MPC approach were RGB, XYZ, rgb, $h_1h_2h_3$, and $l_1l_2l_3$. The use of CIELAB, and CIELUV was not appropriate since the luminance or intensity component would have greatly biased the result. These spaces could be evaluated in the future without the L or I components.

The first experiment consisted in using the color pixel triplet for classification. Experiment two consisted in using a 3-by-3 neighborhood of the color pixel being compared as the actual color thus producing a 27 element vector. The second experiment shows the result of using the implicit smoothing method. There was no explicit smoothing done for the series of experiments conducted here.

The number of classes for each image is different. Usually, one would just select the number of classes for either the k-means or MPC algorithms and run them. However, for increased stability, both algorithms are initiated with a single class to obtain the mean vector of the data in the case of k-means and the first principal component in the case of MPC. This value is quite different for each of the color spaces and it is very important to perform this step in order to obtain meaningful results. After completing the training, a new class prototype is added to the prototypes already in existence. The new prototype is based on one of the class prototypes

already found, and is randomly modified in order to create a new separate class. Each of the iterations performs the clustering step until the number of clusters is reached. If the number of classes is determined to be a power of two (or close to one if this number was large), then the number of class prototypes is doubled instead of adding one new class at each major iteration in order to speed up the training.

The number of training patterns used was 400 times the number of classes for the 1x1 problem and 3600 times the number of classes for the 3x3 clustering. More pixels were used for the 3x3 problem since training in a higher dimensional space requires more data for the data to converge properly. Therefore, the total number of randomly selected training patterns used for an 8 class 3x3 MPC clustering (assuming no overlapping training sets) would have been (8+4+2+1)*3600 = 54,000. That's roughly speaking 19% of the data since the image contains 283162 pixels in total. This would go up to about 40% for the 16 class 3x3 MPC clustering. The number of total training color vectors varied from image to image, but was always less than 50% of all the pixels in the image.

Results for the color clustering approach will be evaluated on Images 1 through 8 described above (see Figure 17). The number of clusters used for segmenting was 16, 16, 8, 14, 3, 5, 4 and 5 for images 1-8 respectively. Quantitative and qualitative results are discussed.

6.3.2 Quantitative Clustering Evaluation

For quantitative algorithm evaluation, the color image segmentation measure proposed by Borsotti et al. [3] was first examined. It was concluded that this Euclidean distance-based measure would not reflect properly the results in this thesis. In this thesis, an alternate way to measure the performance of the above clustering algorithms is proposed.

The new method for evaluating the image segmentation result is based on the idea of maximized pixel consistency within a truthed region [67]. First, it is necessary to manually segment (or truth) the regions of the images that are being examined. In this thesis, only Image 1 is examined in this way since the objects in that scene are easily segmented manually. Figure 18 shows the hand-segmented truth image. Next, the number of different pixel classes in each separate region is counted. Next, the number of pixels corresponding to the most numerous class in each region is added up. These are considered class-consistent pixels since they best represent the truthed region. Then, the second largest number of pixels for each region is calculated, and so forth, up to the number of classes used for the segmentation process.

The quantitative assessment was done on color clustering runs of both the MPC/VA and KM/ED algorithms. The evaluation was done for an 8-class as well as a 16-class clustering to show the degradation that occurs when more classes are added (i.e., the more classes are present the higher the likelihood that a pixel will be closer to a class that it does not belong to).

6.3.3 Qualitative Clustering Evaluation

The bulk of the results will be discussed qualitatively. Figure 28 shows the labels of the details that will be used in the text below to describe the results. The details center on particular features of each of the images. The naming of the details was made consistent with the type of result with which they were associated. The details focus on the following areas:

A – Regions which should be classified as one class

B – Regions where the shadow is mostly or entirely suppressed

C – A region of significantly different saturation sometimes classified as a separate region

D – Regions of similar hue but with different intensity levels

E – Regions affected by inter-reflection

F – Highlight regions (cf. see Section 2.2) which are characterized by having similar physical characteristics as the light source (i.e., highlights appear white due to white light illumination)

G – Small regions with differing hue and saturation from their neighbors

H – Regions of very low intensity

The text will present the author's visual assessment of the results.

6.3.4 Staged Scene Image: Qualitative and Quantitative Results

Image 1 will be first discussed from qualitative and quantitative points of view. This image contains many of the features that could be problematic for color image clustering: falling shadows, object highlights, inter-reflections and superimposed objects of the same hue. The qualitative assessment is given first for each color space and is followed by a quantitative evaluation.

The image segmentation results on Image 1 using the RGB space are shown in Figure 29. It can be immediately noticed that the images showing the results of the MPC/VA algorithm present objects that appear uniformly colored in the scene as single entities compared to the images obtained using the KM/ED algorithm. For example, the blue bottle (i.e., detail A) in Image 1 is seen as one object instead of three. The same is true of the many colored sheets of paper in the background. In addition, most of the shadows have been subsumed into their corresponding backgrounds (e.g., details B) since they result in mostly intensity changes on the background colors.

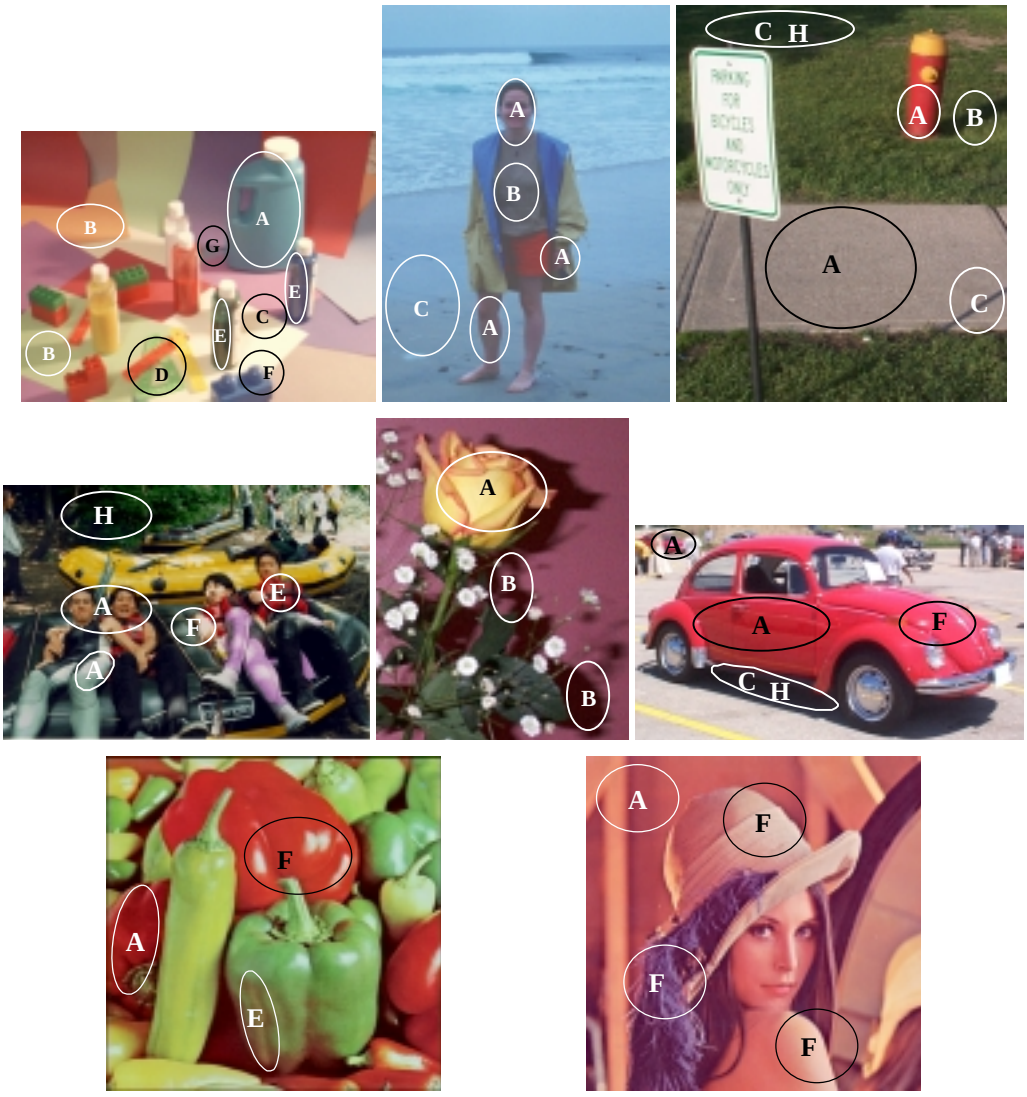


Figure 28: Location of result areas of Images 1-8 discussed in the thesis

These results reflect a tendency of the k-means algorithm to classify image regions based on their intensity levels rather than their hue or saturation. Therefore, any intensity changes on the same object are generally classified as a different object. However, when using the MPC/VA approach, the object's classification is generally based on the object's hue and saturation.

Although many objects do appear whole, some do not. There are, therefore, some limitations to the MPC/VA algorithm. For example, not all shadows have been subsumed into the background objects on which they fall since shadows affect not only the intensity of the color of the object they fall on, but also the color's saturation. In the cases when the change in saturation is high, the shadow will appear as a separate object (e.g., detail C). The same of course happens with KM/ED. It seems that in practice shadows also change the saturation of the color of the matte surface which is not explained by the assumptions used to derive Equation 3.

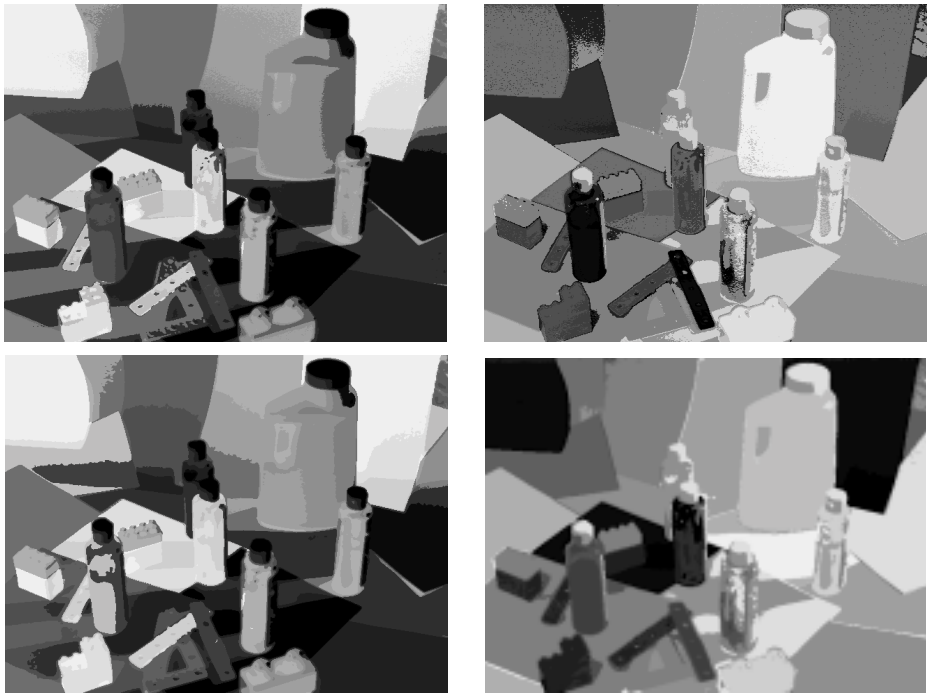


Figure 29: Img 1 RGB results (left to right: KM-1x1, MPC-1x1, KM-3x3, MPC-3x3)

Furthermore, the objects that differ only in intensity from their background will be generally subsumed by the background color. Such is the case with the light green object on the very light green sheet of paper shown in detail D. In the KM/ED case, these objects are well separated in most cases although this is not the case of detail D, but is true of the small bottle of white pain in the background.

Next, effects of inter-reflection (see Section 2.2 for definition) seen in details E affect the segmentation results of both methods although they seem to affect MPC/VA much more than KM/ED. The effect of inter-reflection in Image 1 causes a different color to appear on the glass bottles which leads to these objects not being segmented in their entirety (i.e., each is made up of more than one class if the effect of inter-reflection is present). Finally, highlights (see details F) are clearly visible in all results given that their spectral characteristics are much different from the regions that surround them.

In general, the use of the 3x3 kernel reduces noise and improves qualitative and quantitative (according to the methods presented here) results. However, detail G shows a region that is correctly isolated for the 1x1 case of the MPC/VA, but subsumed into neighboring regions of differing hues in the 3x3 case the MPC/VA (whereas KM/ED separates it correctly). The increase in complexity from a three-dimensional to 27-dimensional space does not result in a noticeable degradation in execution time during the training phase (i.e., computation of the class

prototypes). However, there is an important increase in processing time in the test phase (i.e., when classifying all the pixels in the image).

From a quantitative point of view, the MPC/VA algorithm has a higher pixel consistency rate than the KM/ED approach. Figure 30 and Table 7 show the class frequency distribution in the different color spaces (see Appendix A for detailed numerical values). For example in RGB, 22.2% of the KM/DE results are inconsistent whereas for MPC/VA this figure is 10.6% for the 8-class clustering on a 1x1 neighborhood. When 16 classes were used for clustering colors in Image 1, this resulted in 29.8% of the KM/DE results being inconsistent while this percentage decreased for MPC/VA to 15.2%. When a 3x3 neighborhood was used, a slight improvement was noticed for MPC/VA while there was a 5% degradation in pixel consistency for KM/ED for both the 8-class and 16-class problems. These results agree with the visual evaluation of the images in Figure 29. Furthermore, the degradation from going from an 8-class to a 16-class clustering is expected since there is a higher chance to classify a pixel erroneously.

From the quantitative performance evaluation, it seems that the MPC/VA algorithm shows promise when applied in RGB. The performance measure, however, does not effectively take into account the possibility (which increases as the number of classes decreases) that two adjacent regions may be recognized as the same class (see detail D in Image 1). This is a serious shortcoming when evaluating computer vision systems that should be investigated.

For the XYZ space, the results for KM/ED parallel those of KM/ED in the RGB space (see Figure 31). This is easily explained by the fact that the XYZ space is a linear translation of the RGB space. The MPC results do not seem as good as those in Figure 29. This could be because the intensity component of the XYZ space, color plane Y, is used in the vector angle calculation. For this reason, some shadows that did not appear in Figure 29 appear now (see details B). In addition, the green object of detail D is much better defined. Inter-reflections of detail F are still very prominent (more so than when using RGB). Results for XYZ seem worse than the results for the RGB space. The region of detail G completely disappears when the XYZ space is used regardless of the applied clustering scheme. Quantitatively, the clustering performance in the XYZ space is worse than for RGB (see Figure 30). However, the MPC/VA still has a much better pixel-class consistency rate than KM/ED. This further strengthens the hypothesis that the comparable performance is due to XYZ being a linear translation of RGB.

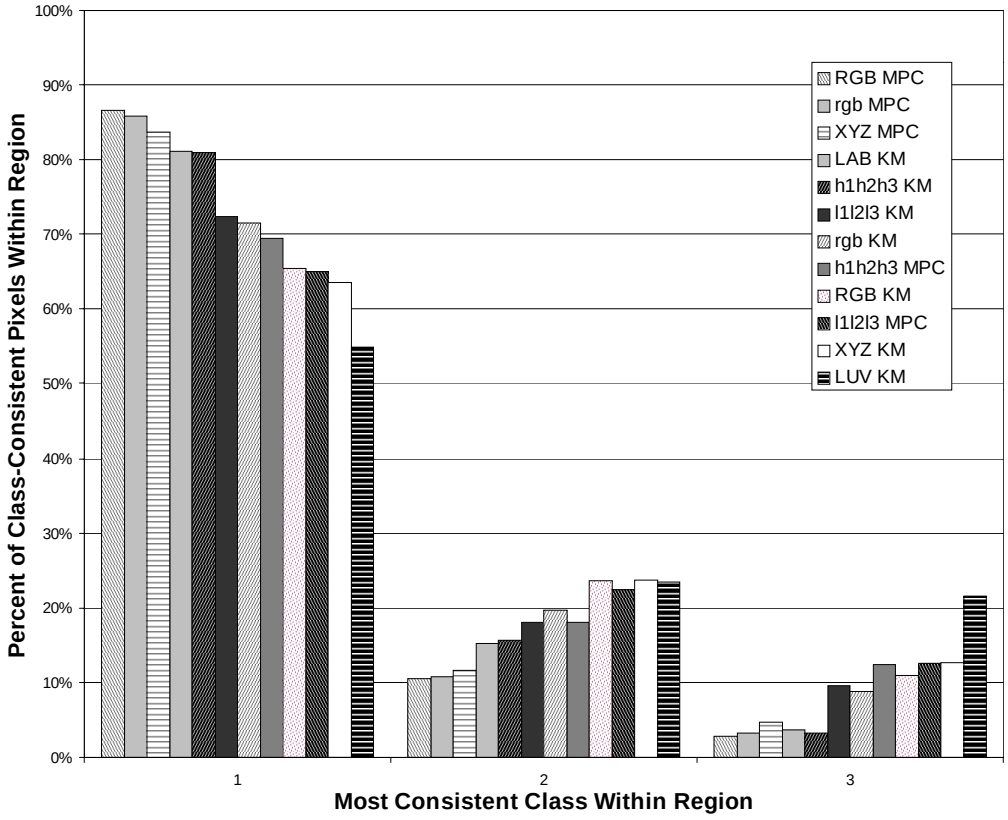


Figure 30: Distribution of most class-consistent pixels

	1	2	3
RGB MPC	86.674%	10.526%	2.800%
rgb MPC	85.854%	10.846%	3.300%
XYZ MPC	83.732%	11.548%	4.720%
LAB KM	81.053%	15.275%	3.673%
h1h2h3 KM	80.999%	15.708%	3.293%
I1I2I3 KM	72.292%	18.079%	9.629%
rgb KM	71.573%	19.698%	8.729%
h1h2h3 MPC	69.498%	18.028%	12.474%
RGB KM	65.406%	23.609%	10.984%
I1I2I3 MPC	65.030%	22.377%	12.593%
XYZ KM	63.543%	23.814%	12.643%
LUV KM	54.978%	23.437%	21.585%

Table 7: Distribution of most class-consistent pixels

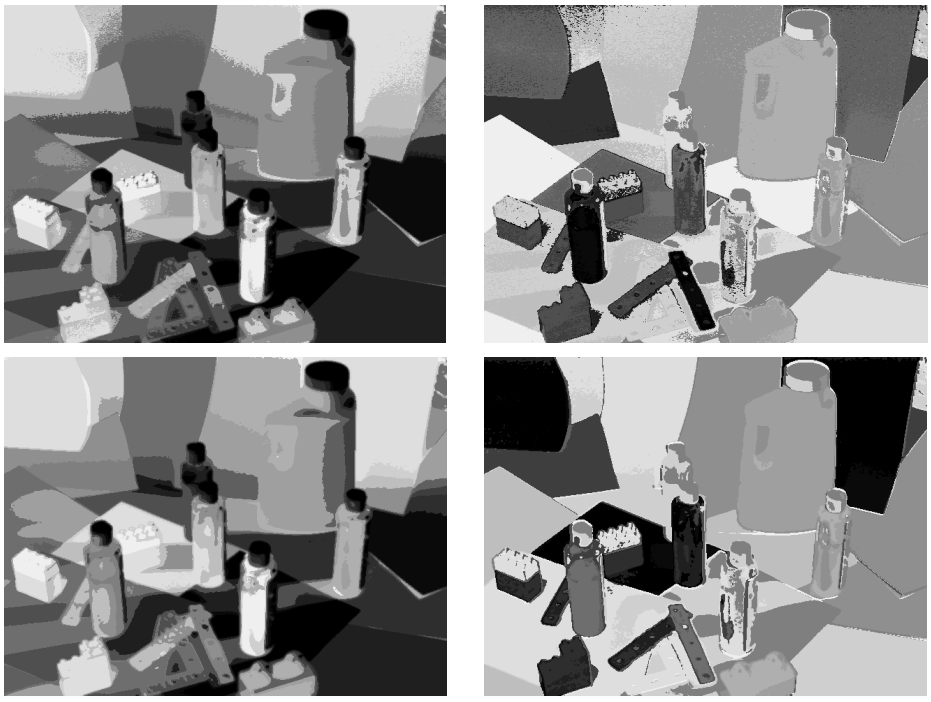


Figure 31: Img 1 XYZ results (left to right: KM-1x1, MPC-1x1, KM-3x3, MPC-3x3)

Figure 32 shows results for the CIELAB and the CIELUV color spaces. Only the KM/ED approach was used here since the intensity component would bias the result of MPC/VA. Both of these results are disappointing. Visually, results in CIELAB look better than the results in CIELUV. This is unexpected given that the CIELUV space was designed as an improvement to the CIELAB space. However, it seems that the results reflect a greater sensitivity of CIELUV to variations in color in hue, saturation and intensity. Since the intensity component is used in the clustering process, it biases the result. The fact that CIELAB and CIELUV are perceptually correct spaces does not seem to result in any benefits for color clustering. The Quantitative evaluation confirms the visual assessment in that it CIELAB seems to have a pixel consistency between 77% and 81% while CIELUV has a deplorable 53%-55% rate. Given that the CILUV space was supposed to have shadow invariant properties, this result is quite poor.

Figure 33 shows results for the rgb (i.e., the normalized RGB) color space. Again, the results seem to parallel the results obtained for the RGB space (cf. Figure 29) with some notable differences. For example, two of the small glass bottles appear whole for the MPC/VA whereas in RGB they appear to be much more fragmented (also for MPC/VA). The region of detail G appears in the KM/ED results and in the 3x3 MPC/VA result. Also, there is more noise in some of the regions due to low values in the rgb space. One of the drawbacks of the rgb color space is that for low intensities it produces color vectors that are meaningless from the color point of view (i.e., the values appear random). When the MPC/VA algorithm is applied, this drawback

might be magnified since the vector angle also produces unreliable measures for low values. However, the qualitative assessment confirms that applying the MPC/VA to the rgb color space is highly effective. Only about 15% of the pixels are inconsistent with MPC/VA while this number jumps to between 29%-31% for KM/ED. Furthermore, the MPC/VA and rgb combination seems to be the only one to come close the MPC/VA performance in RGB. It is surprising that the KM/ED performance is so low given that this space was supposed to exhibit shadow invariant properties. The failure of KM/ED to give comparable performance to the MPC/VA suggests that Equation 2 is not always valid. The shadow invariance built into the MPC/VA seems to be much more effective than that built into the normalized RGB space.

Applying color clustering to the $h_1h_2h_3$ space was surprising (see Figure 34). The KM/ED appears to cluster the image much better than the MPC/VA method. For example, the main body of the blue bottle (i.e., detail A) is only clustered using two classes with KM/ED while four classes are needed when using MPC/VA. In addition, the images for both of the MPC/VA results are very noisy (especially the 1x1 case) which means that 2 or more classes are needed to classify that particular object (see especially right side of image). However, many of the shadows have been subsumed by their respective backgrounds (see details B) when applying both KM/ED and the MPC/VA. Furthermore, it is interesting to note that detail D is classified as a different object in all cases. Also, inter-reflections illustrated by details E are seen in all results. Although this space was shown to be highlight invariant (cf. Section 2.3.8) it seems that highlights in detail F persist. The quantitative evaluation confirms yet again the visual assessment. The number of pixels most consistent with their primary class is around 80% for KM/ED and 58%-69% for MPC/VA.

Finally, Figure 35 shows the results for color clustering using KM/ED and MPC/VA in the $l_1l_2l_3$ color space. It is hard to tell which method clusters the image better given the poor visual appearance of results generated by KM/ED and by MPC/VA. For example, the main body of the blue bottle (i.e., detail A) is only clustered using at least four classes in all cases – many of the illumination changes across the bottle are not clustered into one object. In addition, the segmented images after applying for both methods are very noisy (especially the 1x1 cases) which means that two or more classes are needed to classify each particular object (see especially right and upper-right sides of the image). However, many of the shadows have been subsumed by their respective backgrounds (see details B) although the shadow in detail C persists.

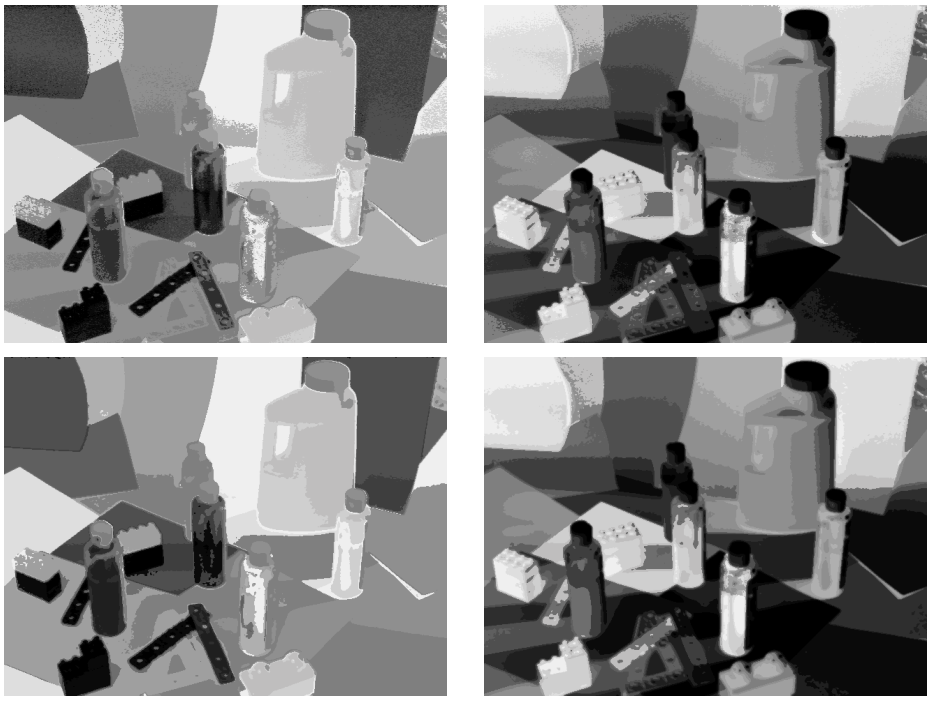


Figure 32: Img 1 CIELAB and CIELUV results (left to right: LABKM-1x1, LUVKM-1x1, LABKM-3x3, LUVKM-3x3)

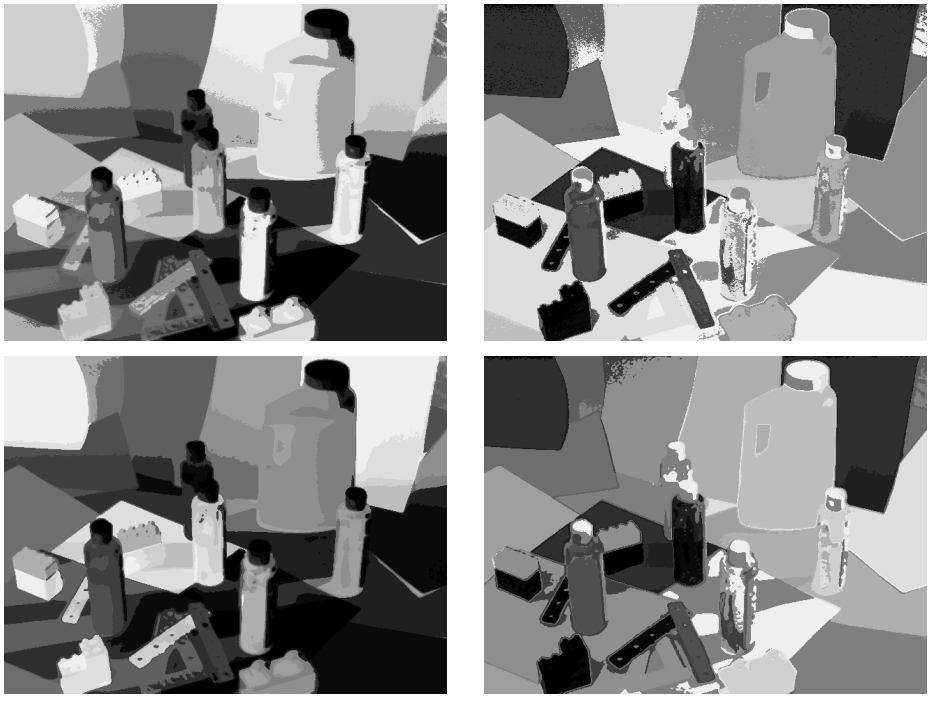


Figure 33: Img 1 rgb results (left to right: KM-1x1, MPC-1x1, KM-3x3, MPC-3x3)

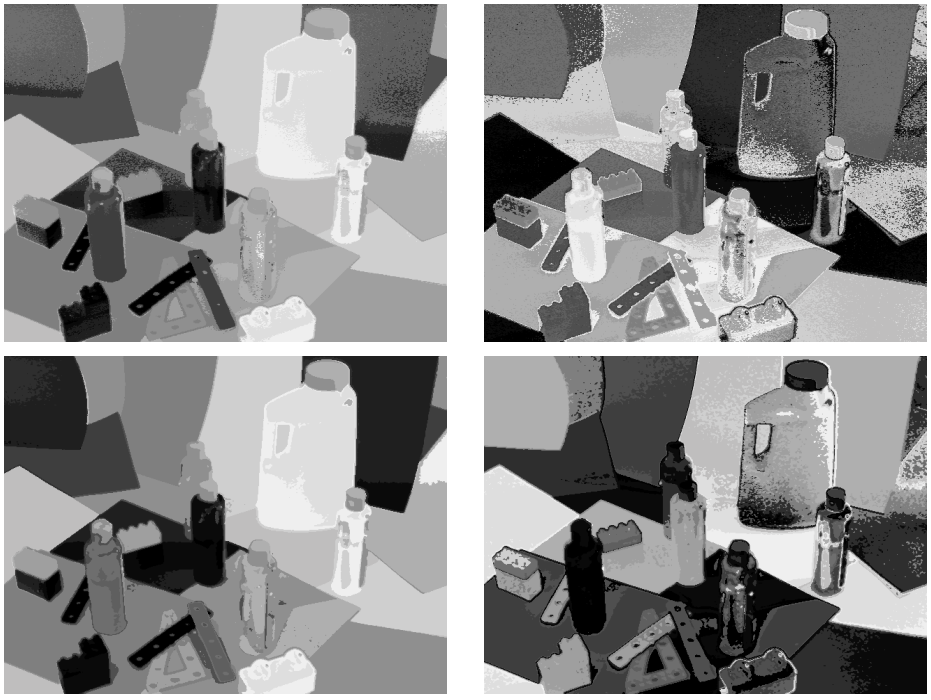


Figure 34: Img 1 $h_1h_2h_3$ results (left to right: KM-1x1, MPC-1x1, KM-3x3, MPC-3x3)

Figure 35: Img 1 $l_1l_2l_3$ results (left to right: KM-1x1, MPC-1x1, KM-3x3, MPC-3x3)

Furthermore, it is interesting to note that detail D is classified as a different object in all cases. Also, inter-reflections illustrated by details E are seen in all results. Although this space was

shown to be invariant in theory to shadows and highlights (cf. Section 2.3.7), the shadows of detail C and highlights in detail F persist. The quantitative evaluation confirms the visual assessment. The clustering in the $l_1l_2l_3$ color space is much worse than in RGB resulting in 28%-39% inconsistent pixels for KM/ED and 35%-37.5% for MPC/VA.

In summary, the three best color space and clustering algorithm combinations appear to be MPC/VA in RGB, followed by MPC/VA in rgb and MPC/VA in XYZ. The best performance for the k-means algorithm was in the CIELAB and the $h_1h_2h_3$ color spaces.

In the next section, several more images will be visually assessed. Quantitative assessment would not be meaningful for those images given the very complex nature of each natural scene.

6.3.5 Natural Scene Images: Qualitative Results

This section will present a qualitative assessment of Images 2-8 in Figure 17. The results obtained in each of the color spaces will be discussed together. Only relevant results for the 3x3 neighborhood will be discussed since the results for the 1x1 neighborhood generally contained more noise and were of worse quality.

First, consider RGB results for Images 2 to 8 shown in Figure 36 (k-means) and Figure 37 (MPC). The skin components in Image 2 (see details A – the hand, the legs and the face) are clustered with the MPC/VA into one class. A similar result is given on Image 4 with MPC/VA although it is not as good as on Image 2. The classification of the skin is not done very well with KM/ED. This is especially seen in details A on Images 2 and 4 (areas that should be classified as the same class but are not). Part of the face in Image 4 near detail E is misclassified as the red lifejacket. Skin detection and classification are still very hard problems today. The MPC/VA offers an alternate way of performing unsupervised classification effectively for skin detection.

Furthermore, areas of constant color shown by details A on Images 2-8 are generally better classified with the MPC/VA. The red Volkswagen in Image 6 seems to have been well segmented by both methods. However, the dark shadow under the car (i.e., details C and H) is not well classified by MPC/VA. This is because MPC/VA fails to classify pixels correctly for low intensities in RGB and low pixel values in general (cf. Section 3.4.1). Dark areas like detail H in Image 3 are also segmented into multiple classes. The same phenomenon responsible for classifying the shadow in detail C in Image 1 (see Figure 1) also happens in Image 2 within the region of sand (i.e., detail C). It is clear that the sand changes in consistency in the area of detail C. Again, this is an important result that should help in improving the theory used in devising spaces and measures invariant to various physics-based lighting effects.

Figure 36: Images 2-8 KM 3x3 RGB results

The MPC/VA segmentation of various objects in Image 3 was very good compared with the KM/ED result. The MPC/VA also clusters the shadow with the background wall for Image 5. However, when the number of classes is increased beyond 3, there does not seem to be any distinction between the performance of KM/ED and MPC/VA for this image. That is, detail B is clustered into one class with the remainder of the shadow.

Figure 37: Images 2-8 MPC-3x3 RGB results

The segmentation of Image 7 is better when using MPC/VA than KM/ED. For example, the shadow area near detail E is clustered (correctly) with the red pepper for MPC/VA, but the KM/ED approach creates one class for this image component. Some of the peppers that appear yellowish are clustered with the greener peppers with MPC/VA. However, KM/ED clusters some the red peppers (see detail A) into a separate class with part of the big green pepper in the center which is undesirable. The segmentation results of Image 8 are quite poor for both algorithms. The MPC/VA algorithm fails since all colors in Image 8 are distributed very close to the gray-axis in the RGB cube. This causes the differences in angle to be small and therefore makes classification difficult. Furthermore, the background has a very similar hue to the face thereby making separation of these two classes difficult. The KM/ED fails since it mostly relies on intensity for segmenting the image. The face has different degrees of shading which causes the intensity to vary across it.

The color clustering results on XYZ and rgb were very similar to those for the RGB space. Therefore, they are not mentioned since the discussion would parallel the one in Section 6.3.4.

Next, some of the color clustering results for Images 2-8 in the CIELAB (see Figure 38) and CIELUV (see Figure 39) spaces are discussed. In general, it appears that CIELAB is better for color clustering than the CIELUV space. Given that CIELUV is based on CIELAB and quantifies better color differences, this is an unexpected result.

One of the better segmentation results for Image 2 appears to be given by applying the KM/ED to CIELAB. This result rivals the result obtained with the 3x3 MPC/VA in RGB; however, the skin regions are not as well segmented when CIELAB is used (notice the different face zones for the in both of the CIELAB results). The other image segmentation results appear worse than those for when using RGB, XYZ or rgb with MPC/VA.

The results of segmenting some of the Images 2-8 in the $h_1h_2h_3$ space are shown in Figure 40 (k-means) and Figure 41 (MPC). The surprising result is that the segmentation with the KM/ED algorithm is much better than that with the MPC/VA. For example, MPC/VA segments the central green pepper into at least four classes (out of a possible four!). The person in Image 2 is generally well segmented by KM/ED except for the skin which is classified into 3 different classes. However, these results are not as good as those obtained with MPC/VA on RGB.

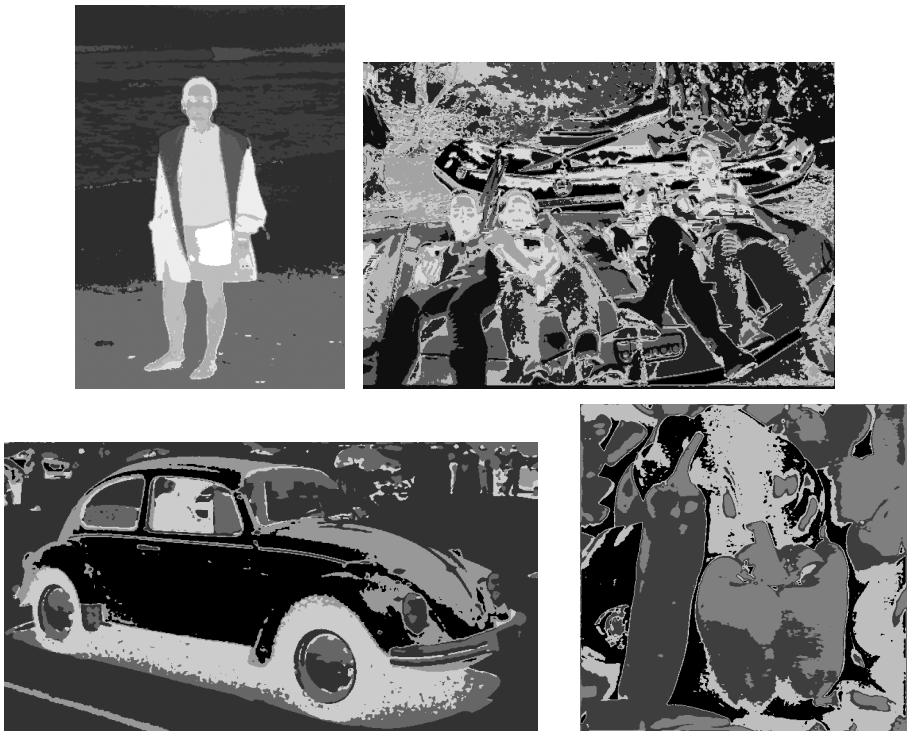


Figure 38: Results on Images 2, 4, 6 & 7 on CIELAB with KM-3x3

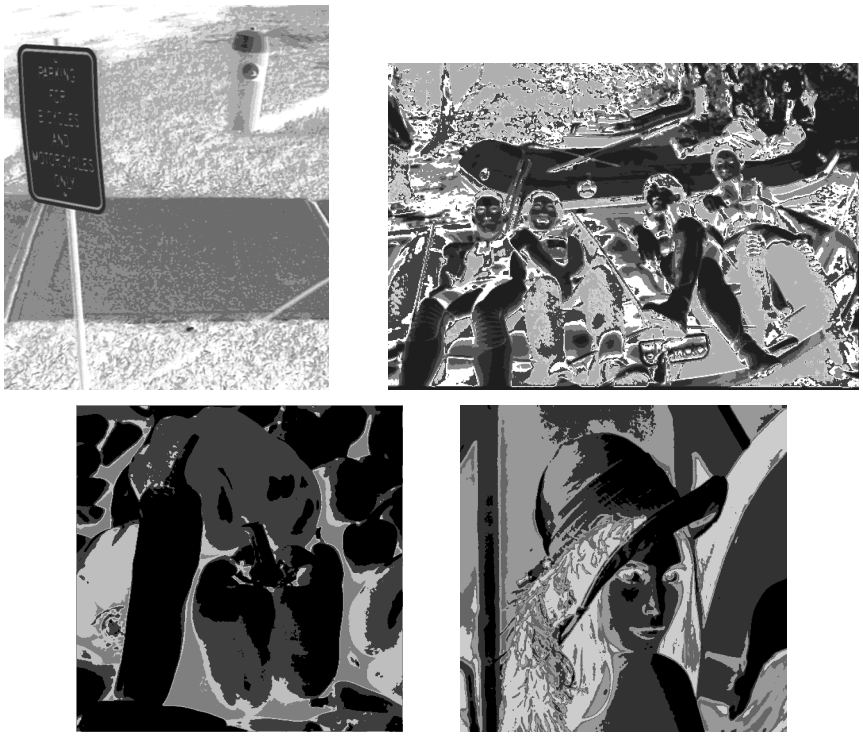


Figure 39: Results for Images 3, 4, 7 & 8 on CIELUV with KM-3x3

Figure 40: Results on Images 2, 3 & 7 on $h_1h_2h_3$ with KM-3x3

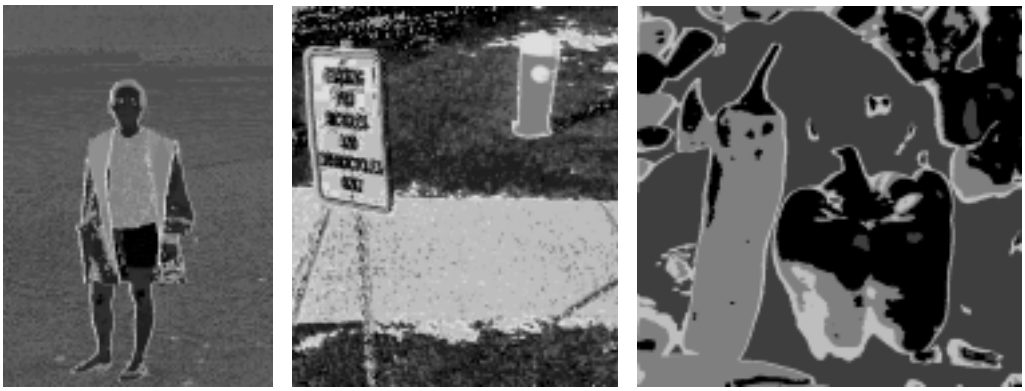


Figure 41: Results on Images 2, 3 & 7 on $h_1h_2h_3$ with MPC 3x3

Figure 42: Results on Image 2 on $l_1l_2l_3$ with KM-3x3 (right) and MPC-3x3 (left)

Finally, the results for $l_1l_2l_3$ were quite poor. The $l_1l_2l_3$ space appears to be very sensitive to small changes in RGB values thereby classifying many visually homogeneous regions into multiple classes. A good example of that can be seen in Figure 42.

6.3.6 Summary of Segmentation Results

The results for image segmentation can be summarized as follows. The MPC/VA algorithm applied to an image in the RGB space yielded the best segmentation results overall on several images with varying characteristics. It is surprising given the results in edge detection that the combination of MPC/VA and one of RGB, XYZ or rgb could yield the best results. The explanation lies within the underlying properties of the combination of color space and similarity measure. Although other spaces combined with the Euclidean distance showed in theory similar properties to the RGB/MPC combination, it appears that in practice the vector angle incorporates certain invariances much better than the transformation of the RGB space into another space (e.g., rgb).

Furthermore, the importance of developing algorithms for specific applications has surfaced yet again. It would not have been possible to predict the results of the color clustering knowing the results of the edge detection. These are in essence two different problems and each needs to have a solution developed with a clear focus on the end goal (i.e., either edge detection or image segmentation).

Chapter 7: Conclusions

7.1 Conclusions

Several conclusions can be drawn based on the research presented in this thesis. They will be grouped by major theme.

7.1.1 Edge Detection

The edge detection algorithms based on the vector angle showed good capabilities in color separation and therefore edge identification in areas where intensities were similar. However, as expected in the areas where hue differences were less significant than intensity differences no edges were generated. These areas were well detected by Euclidean distance-based operators. However, the efforts to combine both measures showed only a small gain with respect to the computational power needed to generate them. This is disappointing in that optimal edge detection was not achieved with this method. The problem most certainly lies in the determination of the trade-off parameter between the two similarity measures. This should be an area of future focus.

The best quantitative and qualitative results were obtained with Euclidean distance-based operators on the $h_1h_2h_3$ space. This is a surprising result as only highlight invariance was deduced from Shafer's model for the $h_1h_2h_3$ space.

7.1.2 Color Clustering

Several conclusions can be made with respect to the color clustering results. The MPC/VA algorithm shows a clear improvement over the KM/ED algorithm from quantitative and qualitative (i.e., visual) points of view. The MPC/VA proved to be the better of the two algorithms within the context of any color space. Furthermore, the top three results were obtained with the MPC/VA algorithm on the RGB, XYZ, and rgb spaces.

Although the quantitative analysis has not yet been perfected, it gives a good indication of the results without having a particular application in mind. This algorithm is a good start for devising an application-independent image segmentation evaluation algorithm.

Certain spaces incorporating physics-based invariances such as $h_1h_2h_3$ and $l_1l_2l_3$ generated poor results with either KM/ED or MPC/VA. This would suggest that certain assumptions could be wrong in the deduction of these properties. This will need to be examined in the future.

The results obtained with KM/ED on the $h_1h_2h_3$ space were much better than the MPC/VA results (although worse than in results obtained in other color spaces). This is unexpected since

the $h_1h_2h_3$ space was not hypothesized to be not intensity-invariant. It was hypothesized that the MPC/VA through the vector angle calculation would introduce an intensity-invariant property into this space. This was not the case and the reasons for this should be studied.

The inclusion of neighboring pixel information in the 3x3 neighborhood implementation was very important in reducing misclassification that occurred in the 1x1 case. This indicates that local information is very important in image segmentation and future research should examine potential ways of incorporating other local information to improve the image segmentation process.

7.1.3 Dichromatic Reflectance Model

Shafer's Dichromatic Reflectance Model has proven to be useful in trying to understand the behavior of the different image processing algorithms in the various spaces studies in this thesis. However, it is felt that the assumptions used in deriving different properties from the model are not always true given the inherently nonlinear nature of the color spaces being studied (especially RGB).

Some of the hypotheses derived based on the model have been shown to be incorrect. For example, the behavior of edge detection algorithms was much better than expected in the $h_1h_2h_3$ color space. Also, the color clustering results in for KM/ED in rgb were much worse than the results for MPC/VA in RGB. This would indicate that the intensity invariance built into the vector angle is much more robust than the intensity invariance built into the normalization of the rgb space. Finally, the results for MPC/VA in both RGB and rgb were not identical. This might be possibly due to a small round-off effect due to the normalization process.

It will be necessary to take a close look at the assumptions in the model to make sure that they are trying to model the physical phenomena as closely as possible. It is possible that the model cannot deal with the data presented in this thesis due to the nonlinear nature of the different color spaces and the non-obvious calculation of the distance between two color points.

7.1.4 Color Spaces

From the color spaces used the $h_1h_2h_3$ color space proved to be most interesting. It negated the hypotheses on its behavior that were generated based on the Dichromatic Reflection Model and gave especially in the edge detection application the best results. Although the color clustering results were not as promising given its very low computational cost, it would be advisable to study this space further.

It can be concluded that some spaces such as XYZ, rgb, $l_1l_2l_3$ and CIELUV are not very good spaces for color image processing applications. This is a surprising conclusion especially for CIELUV which is a perceptually uniform space. Perhaps this color space might be more applicable if the luminance and chrominance components are considered separately.

7.1.5 Computational Speed and Algorithm Performance

The computational cost of combining distances for color edge detection seems to outweigh any positive impact that this combination might have. This is an important practical conclusion as it indicates that such combinations should only be used when they are highly suitable for the application at hand.

The most computationally expensive color spaces showed themselves to be not adequate the applications examined in this thesis. This is an important conclusion as it indicates that research should continue on the RGB space since it good results can be generated based on it as well.

7.2 Future Work

There are numerous possible extensions to this thesis:

1. The study of the vector angle similarity measure itself
2. The extension of the combined Euclidean distance and vector angle similarity measure
3. The possible extensions of other edge detection algorithms to the color image realm and especially the vector angle and combined distance measures
4. The extension of color clustering to a Euclidean distance and vector angle hybrid as well as optimization for various color spaces.
5. The design and implementation of a region growing algorithm based on the vector angle or a combined pixel similarity measure.

Each of these possible extensions will be discussed briefly in the sections below.

7.2.1 The Vector Angle

In this thesis, the vector angle was discussed as the sine of the angle between two vectors. It would be very useful to extend this discussion to include the angle itself and the cosine of the angle. The questions that arise are

1. What is the best representation for the vector angle?
2. Which one is easiest to compute and when?
3. Which one is the most practical and in which situations?

Another issue is the study of the vector angle in spaces other than the RGB space where it could prove to be very useful. One such space was determined to be the $h_1h_2h_3$ color space. However, the current form of the vector angle does not take into account possible negative values in the $h_1h_2h_3$ space. The modified measure could look as follows:

$$D_{VA} = \begin{cases} \|\sin\theta\| & \cos\theta > 0 \\ 2 - \|\sin\theta\| & \cos\theta < 0 \end{cases} \tag{41}$$

Finally, there does not seem to have been any study to evaluate whether the vector angle better reflects perceptual differences than the Euclidean distance in the RGB space or in any other color space. This would be an important addition to the research to determine where the vector angle can best be used.

7.2.2 Combined Similarity Measures

There is much work needed in order to develop practical similarity measures which combine intensity- and hue-based information. This work needs to take on several different facets.

First, answering the question whether the Euclidean distance and vector angle measures are equivalent, i.e., whether one unit in one equals one unit in the other. The simple answer is probably not. Therefore, research is needed in order to develop a framework that can effectively capture the differences between both similarity measures and use them well.

Second, one of the ways this could be done is by determining a statistics-based trade-off function. That is, it would be desirable to assess how the likelihood of detecting an edge with either the Euclidean distance or the vector angle changes as the saturation (or intensity) of the two pixels changes. This will result in a discrete joint probability density function which would uniquely specify the tradeoff parameter between two arbitrary saturation (or intensity) values.

Third, as an alternative to the addition-based combination, a multiplication-based combination of the angular measure and the intensity-based measure could be studied. This combination has been proposed by Androustos et al. for use in color-based image retrieval [1]. A similar measure paralleling the developments of Section 0 would be described as

$$C = 1 - \left[\sqrt{1 - \left(\frac{\vec{v}_1^{\,T}(x,y) \cdot \vec{v}_2(x,y)}{\|\vec{v}_1(x,y)\| \, \|\vec{v}_2(x,y)\|} \right)^2} \right]^{\rho(X_1,X_2)} \left[1 - \frac{\|\vec{v}_1(x,y) - \vec{v}_2(x,y)\|}{\sqrt{3 \cdot 255^2}} \right]^{(1-\rho(X_1,X_2))} \tag{42}$$

The normalization of the Euclidean distance measure is similar to the one proposed in this thesis.

7.2.3 Edge Detection

For edge detection, the main suggested direction of future work is the integration of the vector angle measure into well known edge detectors to improve their performance on color images. The edge detectors in question are especially the Sobel and Canny edge detection operators, as well as the 5x5 median, mean and α-trimmed mean edge detectors [70]. These edge detectors were developed with the Euclidean distance in mind. They have shown good performance in

the presence of noise. However, their performance on a wide range of color images is not known. For example, instead of using the vector mean (i.e., the average of the color vectors) and Euclidean distance as the similarity measure, one could use the principal eigenvector as the class prototype and the vector angle as the similarity measure.

7.2.4 Color Clustering

For color clustering, it is necessary to devise a way to effectively combine the benefits of k-means/Euclidean distance and Mixture of Principal Components / vector angle algorithms. Furthermore, more experimentation is needed with the MPC algorithm to determine practical applications of this work in the color domain.

Another interesting area of research could be the investigation of the optimal color space set-up for clustering with the MPC/VA algorithm. In this thesis, all color spaces have been assumed to have an origin at the (0,0,0) point which allows for the computation of principal eigenvectors from the origin. However, some spaces have points with negative coordinates and this assumption might not be optimal for them.

The problems of skin detection and classification are still challenging today. The MPC/VA offers an alternate way of performing unsupervised classification effectively for skin detection. It appears that the vector angle measure is very good at assessing the distance between pixels in skin regions. This application should be investigated thoroughly.

7.2.5 Region Growing

For region growing, the connected component principles outlined in [62,63] could be extended in a fashion as to exploit the most information in color images. The use of the principal eigenvector of a class instead of the vector mean would be necessary to maximize the results when trying to determine whether a pixel belongs to a growing region. However, computational issues will dictate practical approximations to this preferred approach.

Another region growing algorithm proposed by Hojjatoleslami and Kittler [26] could be extended to color images. A unique feature of the method is that at each step, at most one pixel exhibits the required properties to join the region. In other words, only one pixel at any one time has the desired properties to be added to the growing region. The average contrast of a region is defined as the difference between the average color of the region and the average color of the current boundary (i.e., the pixels that are adjacent to the current region). The peripheral contrast is defined as the difference between the color average of the current internal boundary (i.e., the boundary produced by the set of connected outermost pixels of the current region) and the color average of the current boundary. Again, the principal color defining the region could be determined by the principal eigenvector of that region.

References

1. D. Androutsos, K. N. Plataniotis and A. N. Venetsanopoulos. “Distance Measures for Color Image Retrieval,” *IEEE International Conference on Image Processing*, Chicago, October 1998.

2. D. Aubert and C. Thorpe, *Color Image Processing for Navigation: Two Road Trackers*, Tech. Report CMU-RI-TR-90-09, Robotics Institute, Carnegie Mellon University, April, 1990.

3. M. Borsotti, P. Campadelli, and R. Schettini, "Quantitative evaluation of color image segmentation results," Pattern Recognition Letters, vol. 19, pp. 741-747, 1998.

4. J.K. Bowmaker and H.J. Dartnall. “A Visual pigments of rods and cones in human retina,” *Journal of Physiology*, vol. 298, pp. 501-51, 1980.

5. T. Caelli, and D. Reye, "On the Classification of Image Regions by Color, Texture and Shape," *Pattern Recognition*, vol. 26, no. 4, pp. 461-470, 1993.

6. J. Canny. “A Computational Approach to Edge Detection,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, November 1986.

7. T. Carron and P. Lambert, “Color edge detector using jointly Hue, Saturation and Intensity,” in *Proc. IEEE International Conference on Image Processing*, pp. 977-981, October 1994.

8. T. Carron and P. Lambert, ``Symbolic Fusion of Hue-Chroma-Intensity Features for Region Segmentation," in *Proc. IEEE International Conference on Image Processing*, pp. 971-974, October 1996.

9. M. M. Chang, M. I. Sezan, and A. M. Tekalp, "Adaptive Bayesian segmentation of color images," *Journal of Electronic Imaging*, vol. 3, no. 4, pp. 404-414, October 1994.

10. R. D. Dony, and S. Haykin, "Neural network approaches to image compression," *Proc. IEEE*, vol. 83, pp.288-303, February 1995.

11. R. D. Dony, and S. Haykin, “Multi-class Maximum Entropy Coder (McMEC),” in *IEEE Int'l Conf. Systems, Man, and Cybernetics*, Vancouver, Canada, October 1995.

12. R. D. Dony and S. Wesolkowski, "Edge Detection on Color Images Using RGB Vector Angle," *Proceedings of CCECE 1999*.

13. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Wiley-Interscience, New York, 1973.

14. Adrian Ford and Alan Roberts, Colour Space Conversions, August, 1998. http://www.inforamp.net/~poynton/PDFs/coloureq.pdf

15. H. T. Fung, and K. J. Parker, "Color, complex document segmentation and compression," *Document Recognition IV, Proc. SPIE Vol. 3027*, pp. 180-191, 1997.

16. John Gauch, © University of Kansas, 1997. Source code in C for the color Canny operator: http://www.iv.optica.csic.es/projects/kuim/html/edge/canny.html.

17. Theo Gevers and Arnold W.M. Smeulders, “Color-based object recognition,” *Pattern Recognition*, Vol. 32, pp. 453-464, 1999.

18. Theo Gevers and Arnold W.M. Smeulders, “Reflectance-Based Edge Classification,” *Proceedings of Vision Interface 1999*, Trois-Rivières, Canada, pp. 25-32, May 1999.

19. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Addison-Wesley, 1993.

20. Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision*, vol. 1, Addison-Welsey: Reading, MA, 1992.

21. Mike Heath, Sudeep Sarkar, Thomas Sanocki, and Kevin Bowyer. “A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 12, pp. 1338-1359, December 1997.

22. Mike Heath, Sudeep Sarkar, Thomas Sanocki, and Kevin Bowyer. “Comparison of Edge Detectors: Methodology and Initial Study,” *Computer Vison and Image Understanding*, vol. 69, no. 1, pp. 38-54, January 1998.

23. D. O. Hebb. “The organization of behaviour,” in J.A. Anderson and E. Rosenfeld (eds) *Neuro Computing*, 1988.

24. M. Hedley and H. Yan, ``Segmentation of color images using spatial and color space information," *Journal of Electronic Imaging*, vol. 1, pp. 374-380, October 1992.

25. S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York, NY: Macmillan, 1994.

26. S. A. Hojjatoleslami and J. Kittler. “Region Growing: A New Approach,” *IEEE Trans. on Image Processing*, vol. 7, no. 7, pp. 1079-1084, July 1998.

27. Yingbo Hua, Yong Xiang, Tiangping Chen, Karim Abed-Meraim, and Yongfeng Miao. “Natural Power Method for Fast Subspace tracking,” in Yu-Hen Hu, Jan Larsen, Elisabeth Wilson and Scott Douglas (eds.), Neural Networks for Signal Processing IX, IEEE Press, Piscataway, NJ, 1999, pp. 176-185.

28. Anil K. Jain. *Fundamentals of Digital Image Processing*, Prentice Hall, Inc., Englewood Cliffs, NJ, 1989.

29. Gudrun J. Klinker, Steven A. Shafer and Takeo Kanade, “A Physical Approach to Color Image Understanding,” *International Journal of Computer Vision*, Vol. 4, No. 1, pp. 7-38, 1990.

30. Gudrun J. Klinker, “A physical approach to color image understanding,” Ph.D. thesis, Computer Science Department, Carnegie-Mellon University, May 1988. Available as technical report CMU-CS-88-161.

31. Gudrun J. Klinker, Steven A. Shafer and Takeo Kanade, “The measurement of highlights in color images,” *International Journal of Computer Vision*, Vol. 2, No. 1, pp. 7-32, 1988.

32. F. Kurugöllü, and B. Sankur, “Color Image Segmentation Based on Multithresholding and Fusion,” in *Proc. ICIP*, October 1997. (paper distributed at conference)

33. Levkowitz, Haim, *Color Theory and Modeling for Computer Graphics, Visualization, and Multimedia Applications*. Kluwer Academic Publishers: Norwell MA, USA, 1997.

34. J. Liu and Y.-H. Yang, “Multiresolution Color Image Segmentation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 7, pp. 689-700, July 1994.

35. Q.-T. Luong, "Color in computer vision," *Handbook of Pattern Recognition and Computer Vision*, C.H. Chen, L.F. Pau and P.S.P. Wang editors., pp. 311-368. World Scientific Publishing Company, 1993.

36. A. Moghaddamzadeh, D. Goldman, and N. Bourbakis. “A Fuzzy-like Approach for Smoothing and Edge Detection in Color Images,” *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 12, No. 6, pp. 801-816, 1998.

37. Lawrence O’Gorman and Rangachar Kasturi. *Document Image Analysis*, IEEE Computer Society Press, 1995.

38. Lawrence O’Gorman. “Binarization and Multithresholding of Document Images Using Connectivity,” *CVGIP: Graphical Models and Image Processing*, vol. 56, no. 6, pp. 494-506, November 1994.

39. E. Oja. “A Simplified Neuron Model as a Principal Component Analyzer.” *J. Mathematical Biology*, vol. 15, no. 3, 1982, pp.267-273.

40. S. H. Park, I. D. Yun, and S.U. Lee, "Color Image Segmentation Based on 3-D Clustering: Morphological Approach," *Pattern Recognition*, vol. 31, no. 8, pp. 1061-1076, 1998.

41. Charles Poynton, *A Technical Introduction to Digital Video*. John Wiley & Sons: New York, 1996.

42. Charles Poynton, *Color Frequently Asked Questions*, 1997. http://www.inforamp.net/~poynton/notes/colour_and_gamma/ColorFAQ.html

43. Charles Poynton, *Gamma Frequently Asked Questions*, 1998. http://www.inforamp.net/~poynton/notes/colour_and_gamma/GammaFAQ.html

44. William K. Pratt, *Digital Image Processing, 2nd Edition*. John Wiley & Sons, Inc.: New York, 1991.

45. L. G. Prewitt, "Object Enhancements and Extraction," in *Picture Processing and Psychopictorics*, B. Lipkin and A. Rosenfeld (eds.), Academic Press, New York, NY, 1970, pp. 75-149.

46. B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, 1996.

47. L. G. Roberts, "Machine Perception of Three Dimensional Solids," in *Optical and Electrooptical Information Processing*, J. T. Tippet et al. (eds.), MIT Press, Cambridge, MA, 1965, pp. 159-197.

48. Robert J.Schalkoff, *Pattern Recognition: Statistical, Structural and Neural Approaches.* John Wiley & Sons, Inc., New York, 1992.

49. R. Schettini, "A segmentation algorithm for color images," *Pattern Recognition Letters*, vol. 14, pp. 499-506, June 1993.

50. L. Shafarenko, M. Petrou, and J. Kittler, ``Automatic watershed segmentation of randomly textured color images," *IEEE Trans. on Image Processing*, vol. 6, pp. 1530-1544, November 1997.

51. Steven A. Shafer, "Using color to separate reflection components," *COLOR research and application*, Vol. 10, No. 4, pp. 210-218, 1985. Also available as technical report TR-136, Computer Sciences Department, University of Rochester, NY, April1994.

52. Akira Shiozaki. "Edge Extraction Using Entropy Operator," *Computer Vison, Graphics and Image Processing*, vol. 33, pp. 1-9, 1986.

53. I. E. Sobel, *Camera Models and Machine Perception*, Ph.D. Thesis, Electrical Engineering Department, Stanford University, Stanford, CA, 1970.

54. Nir Sochen and Yehoshua Y. Zeevi, "Representation of Color Images by Manifolds Embedded in Higher Dimensional Non-Euclidean Space," *IEEE International Conference on Image Processing*, Chicago, October 1998.

55. T.G. Stockham, Jr. "Image processing in the context of a visual model," *Proceedings of the IEEE*, vol. 60, pp. 828-842, July 1972.

56. Harro Stokman and Theo Gevers, "Photometric Invariant Region Detectioonin Multi-Spectral Images," *Proceedings of Vision Interface 1999*, pp. 90-96, Trois-Rivières, Canada, 1999.

57. P. Trahanias and A.N. Venetsanopoulos, ``Color edge detection using vector statistics," *IEEE Trans. on Image Processing*, vol. 1, pp. 1-18, July 1992.

58. P. Trahanias and A.N. Venetsanopoulos, ``Color edge detection using vector statistics," *IEEE Trans. on Image Processing*, vol. 2, pp. 259-264, April 1993.

59. A. Tremeau, and N. Borel, "A Region Growing and Merging Algorithm to Color Segmentation," *Pattern Recognition*, vol. 30, no. 7, pp. 1191-1203, 1997.

60. N. Vandenbroucke, L. Macaire, and J.-G. Postaire. “Color Pixel Classification in a Hybrid Color Space,” *IEEE International Conference on Image Processing*, Chicago, October 1998.

61. Brian A. Wandell. *Foundations of Vision*, Sinauer Associates, Inc. Publishers, Sunderland, MA, 1995.

62. Wanzen Wang, Chengyi Sun, and Hongxing Chao. “Color Image Segmentation and Understanding through Connected Components,” *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, pp. 1089-1093, October 1997.

63. Yang Wang and Prabir Bhattacharya. "An Algorithm to Find Parameter Dependent Connected Components of Gray Images," *Machine Graphics and Vision*,, vol. 6, no. 3, 1997, pp. 325-340.

64. G. S. Watson, *Statistics on Spheres*. Wiley-Interscience, 1983.

65. S. Wesolkowski, P. Bowman, and D. Tunnah, ``A high speed color cheque image capture and processing system," in *Proc. IEEE Canadian Conference on Electrical and Computer Engineering*, Waterloo, Canada, pp. 457-460, May 1998.

66. S. Wesolkowski and M.E. Jernigan, "Color Edge Detection in RGB Using Jointly Euclidean Distance and Vector Angle," *Proceedings of IAPR Vision Interface 1999*, Trois-Rivieres, Canada, pp. 9-16, May 1999.

67. S. Wesolkowski, M.E. Jernigan, and R.D. Dony, “Global Color Image Segmentation Strategies: Euclidean Distance vs. Vector Angle,” in Yu-Hen Hu, Jan Larsen, Elisabeth Wilson and Scott Douglas (eds.), *Neural Networks for Signal Processing IX*, IEEE Press, Piscataway, NJ, 1999, pp. 419-428.

68. Woelker, Wolfgang, “Image segmentation based on an adaptive 3D analysis of the CIE-L*a*b* color space,” *Visual Communications and Image Processing ’96*, Proc. SPIE Vol. 2727, pp. 1197-1203, 1996.

69. J. Wu, H. Yan, and A. N. Chalmers, "Color image segmentation using fuzzy clustering and supervised learning," *Journal of Electronic Imaging*, vol. 3, no. 4, pp. 397-403, October 1994.

70. Y. Yang, “Color edge detection and segmentation using vector analysis,” Master's thesis, Electrical and Computer Engineering, University of Toronto, Toronto, Canada, 1995.

71. Silvano Di Zenzo. “A Note on Gradient of Multi-Image,” *Computer Vison, Graphics and Image Processing*, vol. 33, pp. 116-125, 1986.

72. Y. Zhong, K. Karu and A.K. Jain, "Locating Text in Complex Color Images," *Pattern Recognition*, vol. 28, no. 10, pp. 1523-1535, October 1995.

73. J. Zhou, D. Lopresti, and T. Tasdizen, "Finding Text in Color Images," *Document Recognition V, Proc. SPIE Vol. 3305*, pp. 130-140, 1998.

Appendix A: Detailed Color Clustering Consistency Measures

In this appendix, color clustering consistency measure results for Image 1 (see Figure 17) are given. The clustering evaluation measure is explained in Section 6.3.2. Detailed results are presented here for several color spaces: RGB, XYZ, CIELAB, CIELUV, rgb, $h_1h_2h_3$, and $l_1l_2l_3$. The calculations have been done on both 8-class and 16-class clustering results.

RGB
8 classes

K-Means 1x1		K-Means 3x3		MPC 1x1		MPC 3x3	
0	77.766%	0	73.622%	0	89.395%	0	90.712%
1	18.215%	1	22.456%	1	8.898%	1	7.833%
2	2.985%	2	3.166%	2	1.461%	2	1.217%
3	0.866%	3	0.606%	3	0.209%	3	0.217%
4	0.139%	4	0.109%	4	0.032%	4	0.021%
5	0.021%	5	0.036%	5	0.004%	5	0.001%
6	0.004%	6	0.004%	6	0.000%	6	0.000%
7	0.003%	7	0.001%	7	0.000%	7	0.000%

16 classes

K-Means 1x1		K-Means 3x3		MPC 1x1		MPC 3x3	
0	70.157%	0	65.406%	0	84.800%	0	86.674%
1	20.225%	1	23.609%	1	11.530%	1	10.526%
2	6.880%	2	8.501%	2	2.113%	2	1.499%
3	1.748%	3	1.448%	3	0.860%	3	0.708%
4	0.475%	4	0.604%	4	0.403%	4	0.335%
5	0.296%	5	0.243%	5	0.171%	5	0.170%
6	0.131%	6	0.101%	6	0.082%	6	0.061%
7	0.052%	7	0.044%	7	0.027%	7	0.019%
8	0.022%	8	0.025%	8	0.008%	8	0.005%
9	0.007%	9	0.008%	9	0.004%	9	0.001%
10	0.004%	10	0.004%	10	0.000%	10	0.000%
11	0.002%	11	0.002%	11	0.000%	11	0.000%
12	0.000%	12	0.002%	12	0.000%	12	0.000%
13	0.000%	13	0.000%	13	0.000%	13	0.000%
14	0.000%	14	0.000%	14	0.000%	14	0.000%
15	0.000%	15	0.000%	15	0.000%	15	0.000%

XYZ
8 classes

K-Means 1x1		K-Means 3x3		MPC 1x1		MPC 3x3	
0	72.478%	0	67.868%	0	90.965%	0	92.065%
1	20.446%	1	23.668%	1	7.940%	1	6.563%
2	5.299%	2	6.589%	2	0.869%	2	0.866%
3	1.377%	3	1.517%	3	0.185%	3	0.379%
4	0.340%	4	0.289%	4	0.038%	4	0.122%
5	0.047%	5	0.058%	5	0.002%	5	0.006%
6	0.010%	6	0.006%	6	0.000%	6	0.000%
7	0.004%	7	0.003%	7	0.000%	7	0.000%

16 classes

K-Means 1x1		K-Means 3x3		MPC 1x1		MPC 3x3	
0	60.230%	0	63.543%	0	74.713%	0	83.732%
1	24.783%	1	23.814%	1	18.879%	1	11.548%
2	9.813%	2	7.112%	2	4.030%	2	2.924%
3	3.129%	3	3.430%	3	1.340%	3	0.865%
4	1.282%	4	1.260%	4	0.531%	4	0.493%
5	0.432%	5	0.521%	5	0.304%	5	0.239%
6	0.172%	6	0.195%	6	0.120%	6	0.114%
7	0.080%	7	0.083%	7	0.055%	7	0.049%
8	0.035%	8	0.026%	8	0.017%	8	0.028%
9	0.018%	9	0.009%	9	0.007%	9	0.005%
10	0.014%	10	0.004%	10	0.004%	10	0.001%
11	0.005%	11	0.002%	11	0.000%	11	0.000%
12	0.004%	12	0.001%	12	0.000%	12	0.000%
13	0.001%	13	0.000%	13	0.000%	13	0.000%
14	0.001%	14	0.000%	14	0.000%	14	0.000%
15	0.000%	15	0.000%	15	0.000%	15	0.000%

CIELAB and CIELUV

8 classes

LAB K-Means 1x1		LAB K-Means 3x3		LUV K-Means 1x1		LUV K-Means 3x3	
0	88.961%	0	89.330%	0	57.694%	0	61.967%
1	9.028%	1	8.891%	1	27.685%	1	26.019%
2	1.408%	2	1.234%	2	8.881%	2	8.987%
3	0.436%	3	0.428%	3	3.629%	3	2.191%
4	0.154%	4	0.115%	4	1.282%	4	0.573%
5	0.012%	5	0.002%	5	0.546%	5	0.214%
6	0.001%	6	0.000%	6	0.197%	6	0.044%
7	0.000%	7	0.000%	7	0.086%	7	0.006%

16 classes

LAB K-Means 1x1		LAB K-Means		LUV K-Means 1x1		LUV K-Means 3x3	
0	77.252%	0	81.053%	0	53.056%	0	54.978%
1	17.194%	1	15.275%	1	24.169%	1	23.437%
2	4.152%	2	2.396%	2	11.878%	2	11.909%
3	0.739%	3	0.701%	3	6.505%	3	5.053%
4	0.319%	4	0.328%	4	2.673%	4	2.921%
5	0.181%	5	0.133%	5	0.980%	5	0.815%
6	0.117%	6	0.076%	6	0.303%	6	0.414%
7	0.030%	7	0.029%	7	0.188%	7	0.222%
8	0.013%	8	0.008%	8	0.108%	8	0.142%
9	0.002%	9	0.001%	9	0.067%	9	0.059%
10	0.001%	10	0.000%	10	0.043%	10	0.028%
11	0.000%	11	0.000%	11	0.018%	11	0.008%
12	0.000%	12	0.000%	12	0.004%	12	0.008%
13	0.000%	13	0.000%	13	0.003%	13	0.004%
14	0.000%	14	0.000%	14	0.003%	14	0.003%
15	0.000%	15	0.000%	15	0.000%	15	0.002%

Normalized RGB: rgb

8 classes

K-Means 1x1		K-Means 3x3		MPC 1x1		MPC 3x3	
0	74.514%	0	77.602%	0	92.390%	0	92.047%
1	20.541%	1	18.704%	1	5.891%	1	6.969%
2	3.978%	2	2.768%	2	1.193%	2	0.654%
3	0.792%	3	0.784%	3	0.382%	3	0.263%
4	0.149%	4	0.127%	4	0.128%	4	0.059%
5	0.015%	5	0.011%	5	0.014%	5	0.008%
6	0.010%	6	0.004%	6	0.001%	6	0.000%
7	0.002%	7	0.000%	7	0.000%	7	0.000%

16 classes

K-Means 1x1		K-Means 3x3		MPC 1x1		MPC 3x3	
0	69.210%	0	71.573%	0	85.056%	0	85.854%
1	21.381%	1	19.698%	1	11.447%	1	10.846%
2	5.994%	2	6.175%	2	1.793%	2	2.025%
3	2.280%	3	1.506%	3	0.938%	3	0.664%
4	0.649%	4	0.581%	4	0.495%	4	0.303%
5	0.279%	5	0.242%	5	0.176%	5	0.203%
6	0.119%	6	0.132%	6	0.072%	6	0.077%
7	0.043%	7	0.059%	7	0.014%	7	0.020%
8	0.027%	8	0.023%	8	0.006%	8	0.007%
9	0.010%	9	0.006%	9	0.001%	9	0.002%
10	0.004%	10	0.003%	10	0.000%	10	0.000%
11	0.002%	11	0.002%	11	0.000%	11	0.000%
12	0.001%	12	0.000%	12	0.000%	12	0.000%
13	0.000%	13	0.000%	13	0.000%	13	0.000%
14	0.000%	14	0.000%	14	0.000%	14	0.000%
15	0.000%	15	0.000%	15	0.000%	15	0.000%

$h_1h_2h_3$

8 classes

K-Means 1x1		K-Means 3x3		MPC 1x1		MPC 3x3	
0	93.499%	0	94.171%	0	73.212%	0	79.329%
1	5.691%	1	5.089%	1	19.064%	1	13.475%
2	0.619%	2	0.584%	2	4.502%	2	4.275%
3	0.135%	3	0.127%	3	1.671%	3	1.531%
4	0.055%	4	0.029%	4	0.747%	4	0.794%
5	0.000%	5	0.000%	5	0.505%	5	0.356%
6	0.000%	6	0.000%	6	0.221%	6	0.165%
7	0.000%	7	0.000%	7	0.077%	7	0.076%

16 classes

K-Means 1x1		K-Means 3x3		MPC 1x1		MPC 3x3	
0	80.011%	0	80.999%	0	57.812%	0	69.498%
1	15.999%	1	15.708%	1	26.035%	1	18.028%
2	2.676%	2	2.388%	2	6.810%	2	5.365%
3	0.860%	3	0.564%	3	3.547%	3	2.728%
4	0.292%	4	0.192%	4	2.174%	4	1.489%
5	0.106%	5	0.085%	5	1.211%	5	0.857%
6	0.041%	6	0.035%	6	0.682%	6	0.594%
7	0.011%	7	0.022%	7	0.467%	7	0.500%
8	0.003%	8	0.006%	8	0.358%	8	0.299%
9	0.001%	9	0.001%	9	0.256%	9	0.237%
10	0.000%	10	0.000%	10	0.213%	10	0.185%
11	0.000%	11	0.000%	11	0.179%	11	0.082%
12	0.000%	12	0.000%	12	0.119%	12	0.045%
13	0.000%	13	0.000%	13	0.062%	13	0.034%
14	0.000%	14	0.000%	14	0.043%	14	0.030%
15	0.000%	15	0.000%	15	0.034%	15	0.026%

$I_1I_2I_3$

8 classes

K-Means 1x1		K-Means		MPC 1x1		MPC 3x3	
0	82.081%	0	83.400%	0	77.624%	0	81.874%
1	12.763%	1	12.081%	1	15.985%	1	12.674%
2	2.660%	2	2.595%	2	3.912%	2	3.254%
3	1.183%	3	0.911%	3	1.137%	3	1.128%
4	0.663%	4	0.477%	4	0.670%	4	0.537%
5	0.298%	5	0.307%	5	0.428%	5	0.370%
6	0.264%	6	0.174%	6	0.144%	6	0.114%
7	0.088%	7	0.053%	7	0.101%	7	0.049%

16 classes

K-Means 1x1		K-Means 3x3		MPC 1x1		MPC 3x3	
0	60.793%	0	72.292%	0	62.501%	0	65.030%
1	24.300%	1	18.079%	1	22.862%	1	22.377%
2	6.780%	2	4.891%	2	7.331%	2	6.268%
3	3.659%	3	1.999%	3	2.918%	3	2.480%
4	1.490%	4	0.977%	4	1.419%	4	1.281%
5	0.952%	5	0.596%	5	0.857%	5	0.803%
6	0.624%	6	0.474%	6	0.604%	6	0.586%
7	0.370%	7	0.284%	7	0.417%	7	0.334%
8	0.254%	8	0.171%	8	0.275%	8	0.243%
9	0.215%	9	0.092%	9	0.240%	9	0.177%
10	0.166%	10	0.051%	10	0.165%	10	0.145%
11	0.140%	11	0.038%	11	0.151%	11	0.121%
12	0.119%	12	0.023%	12	0.113%	12	0.067%
13	0.072%	13	0.017%	13	0.070%	13	0.045%
14	0.043%	14	0.010%	14	0.053%	14	0.027%
15	0.022%	15	0.006%	15	0.024%	15	0.015%