Finding Near Optimum Colour Classifiers : Genetic Algorithm-Assisted Fuzzy Colour Contrast Fusion using Variable Colour Depth

A thesis presented to the

Institute of Information and Mathematical Sciences

in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science

at

Massey University, Albany, Auckland, New Zealand

By

Heesang Shin

May 2009

Abstract

This thesis presents a complete self-calibrating illumination intensity-invariant colour classification system. We extend a novel fuzzy colour processing technique called Fuzzy Colour Contrast Fusion (FCCF) by combining it with a Heuristic-assisted Genetic Algorithm (HAGA) for automatic fine-tuning of colour descriptors. Furthermore, we have improved FCCF's efficiency by processing colour channels at varying colour depths in search for the optimal ones. In line with this, we introduce a reduced colour depth representation of a colour image while maintaining efficient colour sensitivity that suffices for accurate real-time colour-based object recognition. We call the algorithm Variable Colour Depth (VCD) and we propose a technique for building and searching a VCD look-up table (LUT). The first part of this work investigates the effects of applying fuzzy colour contrast rules to varying colour depths as we extract the optimal rule combination for any given target colour exposed under changing illumination intensities. The second part introduces the HAGA-based parameter-optimisation for automatically constructing accurate colour classifiers. Our results show that for all cases, the VCD algorithm, combined with HAGA for parameter optimisation improve colour classification via a pie-slice colour classifier.For 6 different target colours, the hybrid algorithm was able to yield 17.63% higher overall accuracy as compared to the pure fuzzy approach. Furthermore, it was able to reduce LUT storage space by 78.06% as compared to the full-colour depth LUT.

Preface

Some merits of this work has already been recognised, published and submitted.

Lecture Notes in Computer Science, Springer-Verlag (accepted, to appear in 2009) : Variable Colour Depth Look-Up Table Based on Fuzzy Colour Processing, Heesang Shin and Napoleon H. Reyes, In Proceedings of ICONIP 2008

Memetic Computing Journal, Springer (submitted in 2009) : Finding Near Optimum Colour Classifiers : Genetic Algorithm-Assisted Fuzzy Colour Contrast Fusion using Variable Colour Depth, Heesang Shin and Napoleon H. Reyes

Acknowledgements

First , I would like to thank Dr. Napoleon Hamoay Reyes, without his insight and guidance I wouldn't finish this thesis. That's why I used pronoun we instead of I, he deserves it. I also equally thanks to my wife Kristen KyungEun, it wouldn't appropriate to list just number of things to express her total support. Finally I dedicate this thesis to my elder son Daniel JinKyu, without him I wouldn't able to return to study after long years in industry.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Research Description

1.1 Overview of the Current State of Technology

Colours depicting a moving object, as captured by a camera change under spatially varying illuminations. In addition, confounding effects due to the spectral reflectance characteristic of the object, the spectral power distribution of the illuminant [13] and sensitivity of the camera make the colour classification task very difficult [10]. On the contrary, our model, the human visual system is able to recognise the colours of objects irrespective of the light used to illuminate them. This ability is called colour constancy and is a result of human evolution that adjusts the white balance dynamically as lighting changes [14]. Unlike other visual properties like shape and size, colours are considered to be view invariant and largely independent of resolution [15]. This makes colours a highly valuable property for object recognition, but colours are only stable under constant illumination. This raises difficulty and opportunity for computer vision research, and in particular, colour classification. The colour classification process we refer to here pertains to correctly distinguishing colours belonging to the same object as it moves across the scene, with the presence of other similar colours. Although there are many researches about colour classification, only few researches successfully classify colours under varying illuminations [10, 12, 11, 16, 5, 7, 1].

FCCF(Fuzzy Colour Contrast Fusion) is one of the promising algorithms that offers a solution to classifying colours in real-time, under spatially varying illuminations and presence of other similar colours that should be effectively distinguished [1]. FCCF embodies colour constancy by employing colour contrast operations on each colour channel independently using fuzzy logic. However, FCCF requires a multitude of colour classification parameters for each target colour object. These parameters are divided into two parts, the pie-slice decision region (classification angles and radii), and the colour contrast operation rules. Finding the pie-slice classification region originally depends on manual calibration and is a painstaking process. In [11], a motion-based colour learning technique was proposed to extract the pie-slice classification region, but the extraction of the optimal colour contrast operation rules was performed only through a brute-force search method. What is desirable is an automatic colour classification system that searches for the pie-slice classification region and colour contrast rules more effectively. The system should learn the illumination conditions of the scene and classify the target colours without any further human intervention. Furthermore, improvement of the algorithm's accuracy and reduction of memory requirements are worth researching about. These desirable features are studied and dealt with in this thesis.

1.2 Research Objectives

1.2.1 General Objective

To develop an automatic colour classification system which examines input scenes with initial target colours positioned on various illumination points to generate the optimal colour classification parameters. The target colours should cover the brightest and dimmest positions of the scene as well as a number of them in between illuminations. The optimal colour classification parameters should be effective until the source of illumination conditions completely change.

1.2.2 Specific Objectives

1. To investigate how the look-up table's memory space consumption could be reduced without sacrificing colour classification accuracy and speed.

2. To investigate how the colour pixels are affected with reduced colour depth (e.g. 5-bits for red, 5-bits for green, 5-bits for blue).

3. To investigate the colour discriminability of FCCF under varying colour depth.

4. To devise an appropriate chromosome representation that will incorporate all the colour classifier properties. The number of decimal places of each colour feature should suffice the required colour classification accuracy.

5. To develop a platform independent FCCF colour classification parameter search engine using Genetic Algorithm to find the optimal colour classification parameters.

6. To compare the proposed algorithms with the latest advances in FCCF.

1.3 Scope and Limitations of Research

This research limits colour classification to colours definable in the modified rg-chromaticity colour space with a pie-slice decision region. This excludes colours such as the different levels of gray, brown and gold colours.

1.4 Research Methodology

The bulk of the research will focus on the increase of colour classification accuracy by searching for the optimal colour classification parameters and revise the FCCF algorithm. Genetic Algorithm will be utilised to search for the optimal colour classification parameters in a large search space. The FCCF algorithm will be revised in terms of logic optimisation and memory consumption.

1.5 Structure of the Thesis

This thesis begins with a theoretical framework covering the concept of colour, colour spaces and colour acquisition as well as colour representation followed by a review of related literatures. Consequently, we introduce Variable Colour Depth look-up table and Fuzzy-Genetic colour classification which is followed by a series of experiments of algorithms. Finally we will discuss the results of the experiments and conclude this thesis.

Additionally as part of the appendix, we attach a list of software used in the development of the proposed system as well as the source code of the proposed system.

Chapter 2

Theoretical Framework

2.1 Colour

Koschan and Abidi defines colour as a perceived phenomenon and not a physical dimension like length or temperature, although the electromagnetic radiation of the visible wavelength spectrum is measurable as a physical quantity [17]. The visible wavelength spectrum is found when Sir Isaac Newton observed that a glass prism could decompose a beam of sunlight into continuous spectrum of colours ranging from violet at one end to red at the other in 1666. We perceive colours of an object mainly because an object reflects light of certain wavelengths and absorbs the rest of the other wavelengths in the visible spectrum; this is called the spectral reflectance characteristic. If the incident light on the object is not white then the colour might look different even for the same object. Other contributing factors also affect colour, such as potentially on the angles of illumination and viewing and some objects not only reflect light, but also transmit light or emit light themselves. When light arrives on our eyes, it projected on the retina through the lenses. The retina is filled with photoreceptors which consist of rods and three types of cones. Rods are sensitive to brightness, while and cones have types sensitive to the short, medium and long wavelengths which correspond closely to blue, green and red.

Figure 1 shows that the visible spectrum resides between the Ultra Violet and Infra Red. Figure 2 illustrates how colour perceived from object to human. When a light arrives at leaf, some wavelengths are reflected at surface and others are absorbed. The reflected wavelengths are determine colour of the object to human, in this case green is reflected colour. Figure 3 shows schematics of the human eye and cross section view of retina where colour is initially perceived.

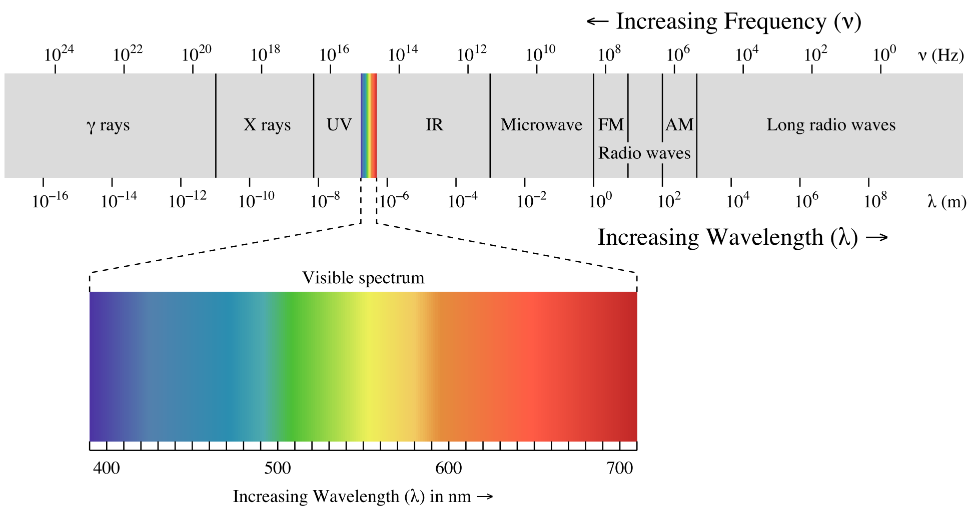

Figure 1: Electromagnetic spectrum with Light Highlighted *Picture created by Philip Ronan from Wikipedia*

Colour perception is not done in colour receptors in retina but concludes in brain, colour constancy is good example that human vision system is able to recognise the colour of objects irrespective of the light used to illuminate the objects. Although human colour constancy ability enable to recognise white paper under reddish tungsten lighting, greenish fluorescent lighting or bluish in daylight shadows, also causes optical illusion called same colour illusion[2] that when interpreted as a 3-dimensional scene, human visual system immediately estimates a lighting vector and uses this to judge the property of the material. Figure 4 shows example of same colour illusion.

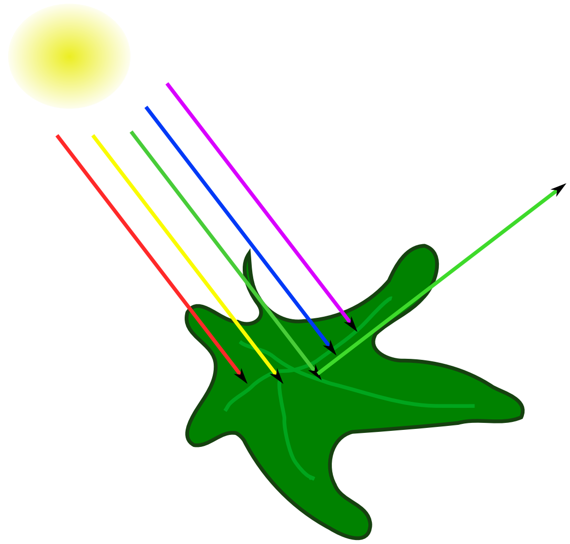

Figure 2: Leaf Reflects Green Wavelength on The Surface to be Perceived

2.2 Colour Space Models

Gonzalez and Woods describe a colour model is a specification of a coordinate system and a subspace within that system where each colour is represented by a single point[18]. There are many number of colour space models are introduced to facilitate specific needs to acquire, reproduce and processing the colour image such as RGB (red,green,blue) , CMY (cyan,magenta,yellow) and HSI (hue,saturation,intensity)

2.2.1 CIE XYZ Colour Space and CIE 1931 Chromaticity Diagram

Wright and Guild derived CIE XYZ colour space from a series of experiments of visible spectrum in the late 1920s and developed diagram called CIE xy (also called CIE 1931) Chromaticity Diagram. It is one of the first mathematically defined colour space created by the International Commission on Illumination (CIE) in 1931.

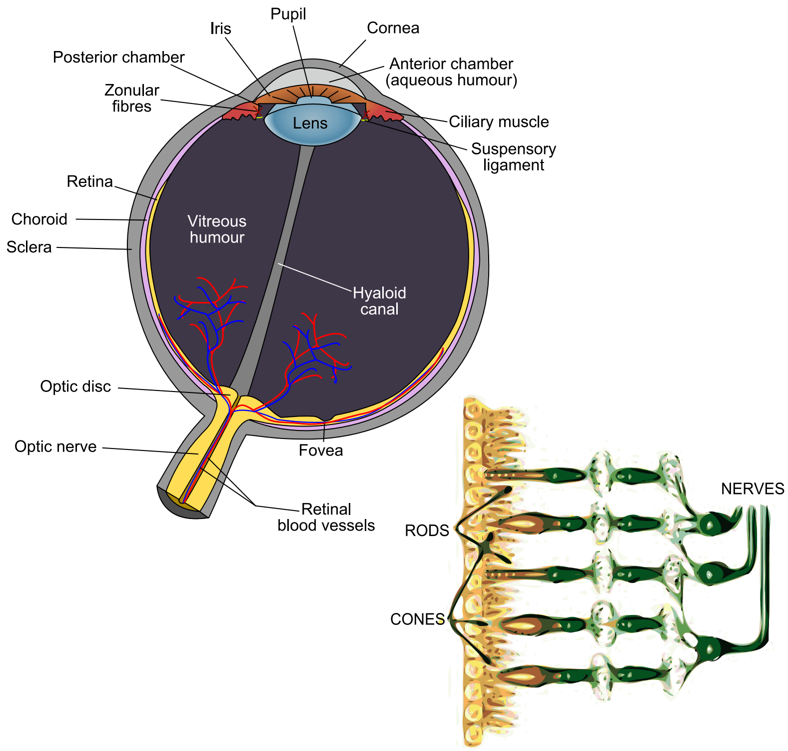

Figure 3: Schematic Diagram of the Human Eye and Cross Section View of Retina. *Schematic created by Rhcastilhos from Wikipedia*

The CIE XYZ colour space is based on human vision system that the human perceiving colour through three types colour receptors known as cone cells which produce a colour sensation. The cones are labelled short(S), middle(M), and long(L) cone types for wavelengths of the peaks of their spectral sensitivities. The tristimulus values of S, M, and L where wavelengths peak roughly at red, green, and blue are corresponds to the CIE XYZ colour space, X, Y, and Z respectively. The component X,Y and Z derived from colour matching functions which based on standard observer which taken to be the chromatic response of the average human viewing through a 2° angle, due to the belief that the colour-sensitive cones resided within a 2° arc of the fovea. [3] Figure 5 shows normalised human cone response curve on wavelengths and corresponding CIE standard observer colour-matching functions.

The CIE xy Chromaticity Diagram is a two-dimensional plot defining colour of CIE XYZ colour space shown in figure 6. In order to plot three-dimensional figure

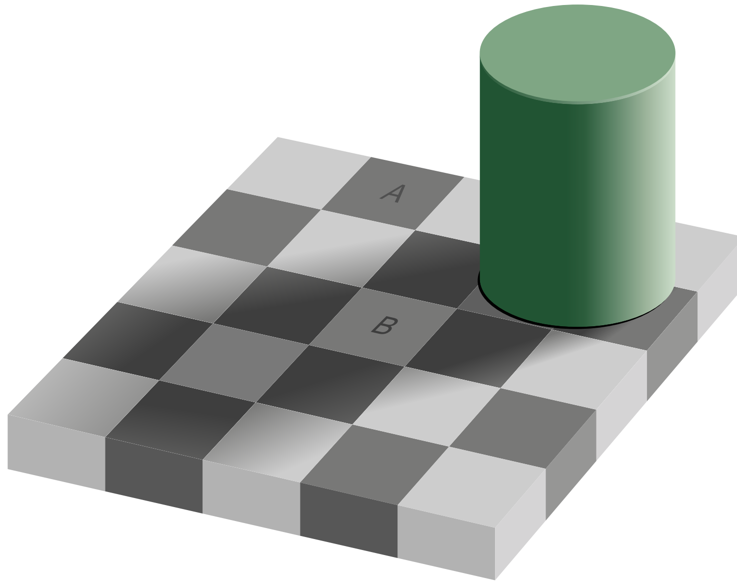

Figure 4: An Optical Illusion. Square A is Exactly the Same Shade of Grey as Square B. *Picture created by Adrian Pingstone, based on the original created by Edward H. Adelson [2]*

of CIE XYZ colour space to two-dimensional space, CIE xyY colour space is derived from CIE XYZ colour space. In CIE xyY colour space, the chromaticity of a colour was then specified by the two derived parameters x and y, two of the three normalised values which are functions of all three tristimulus values X, Y, and Z. Equation 1 shows derivation of x, y components and Equation 2 shows how original value of X and Y calculated back.

$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z} = 1 - x - y \tag{1}$$

$$X = \frac{Y}{y}x$$

$$Z = \frac{Y}{y}(1 - x - y) \tag{2}$$

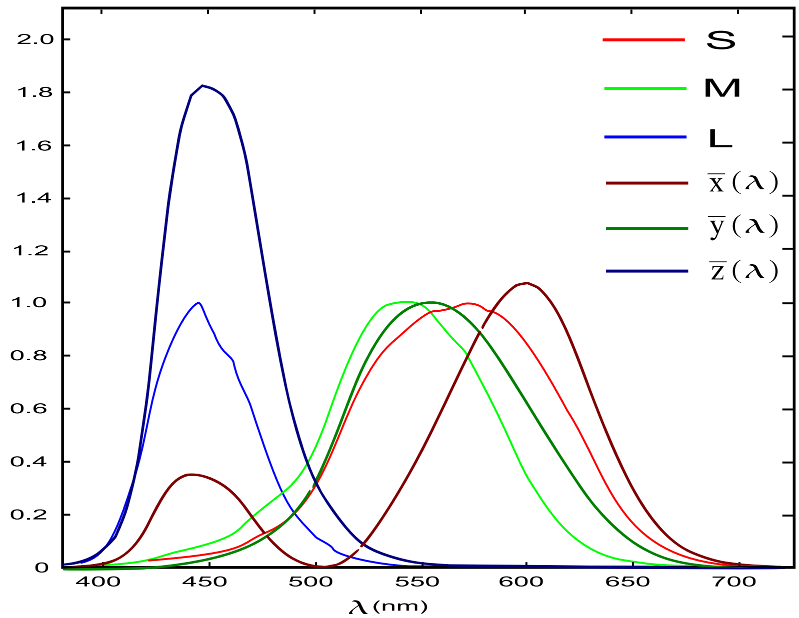

Figure 5: Simplified Human Cone Response Curve and Corresponding CIE XYZ Colour Matching Function

In the Figure 6, the outer curved boundary is the spectral (or monochromatic) locus, with wavelengths shown in nanometres. However, the colours depicted in the diagram is depend on the colour space of the device on which is viewed the image, and no device has a gamut large enough to present an accurate representation of the chromaticity at every position.

2.2.2 RGB Colour Model

RGB colour space model consists of three primary colours of lights that are red, green and blue. Each colour assigned to its independent component(or channel) and represent its saturation level. A colour in RGB colour space defined triplet of each component value like *1r0g0b* which is most pure reddish colour in RGB colour space saturation level normalised to the range [0,1]. In RGB colour space, black is defined minimum level of red, green and blue and white defined maximum level of red, green and blue. The RGB colour space commonly described as a shape of cube that each axis assign to red, green and blue. Figure 7 and 8 shows the schematic of the RGB colour cube. The RGB colour cube shows three corners on farthest

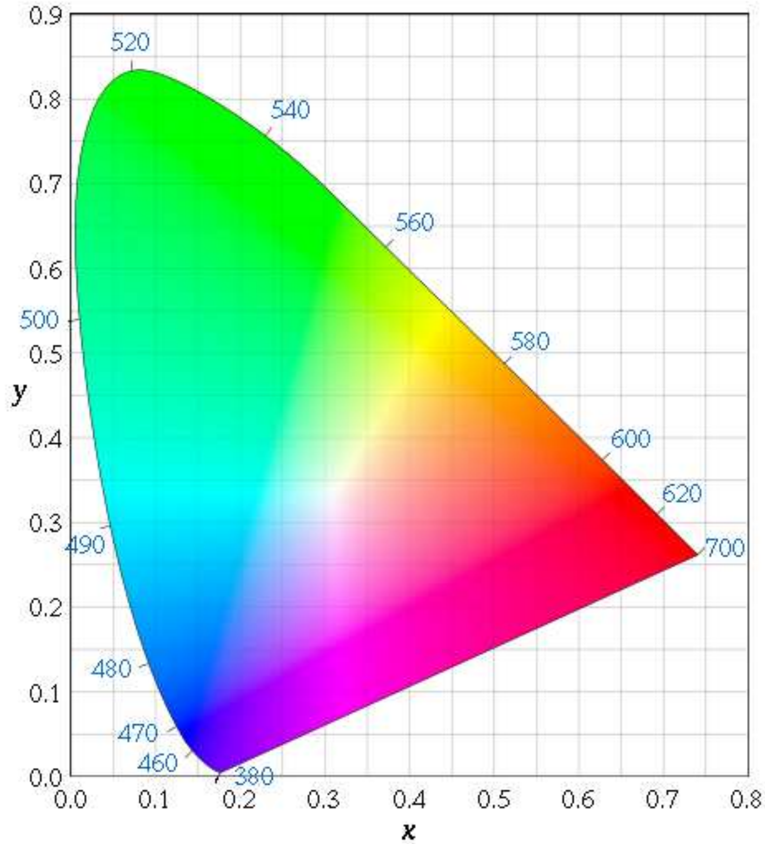

Figure 6: The CIE 1931 Colour Space Chromaticity Diagram. *Picture created by Sakurambo from Wikipedia, based on the original created by CIE [3]*

of each axis for red, green and blue and three other corners of where two axis are met for magenta (blue,red), cyan (blue,green) and yellow (red,green). Finally two corners of where three axis are met for black where all the axis are at minimum and white where all the axis are at maximum. Since RGB colour space based on colours of lights, it commonly used in display devices which output is based on emitting lights.

2.2.3 CMY and CMYK Colour Models

CMY colour space model is similar to RGB colour space model in many aspects. CMY colour space model consists of three primary colours of pigments that are

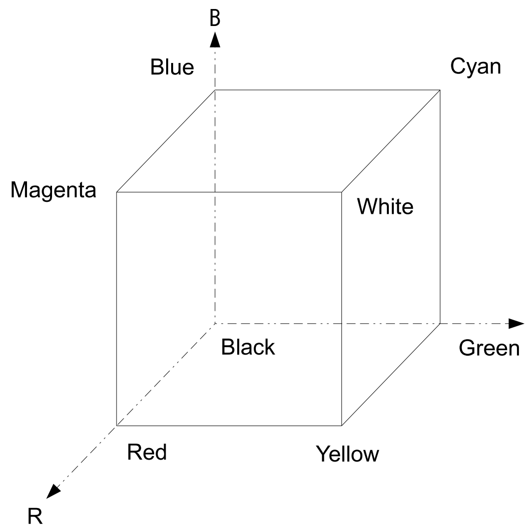

Figure 7: Schematic of the RGB Colour Cube

cyan, magenta and yellow. In CMY colour space, white and black defined exact opposite of in RGB colour space. White is defined when all the component colours at minimum level as like white paper without any ink on it and black is defined when all the component colours at maximum level.
Converting RGB to CMY colour space easily done by simple operation as Equation 3 assume that all colour values have been normalised to the range [0,1].

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3}$$

Although CMY colour space is suitable colour space for printing devices which output is based on pigments, CMY colour space is rarely used as original form because mixture of cyan, magenta and yellow pigments commonly resulted in mucky black on the paper. Instead of CMY colour space, printing devices commonly uses CMYK colour space which is based on CMY colour space with addition of black as forth colour pigment, which K stands for key or black.

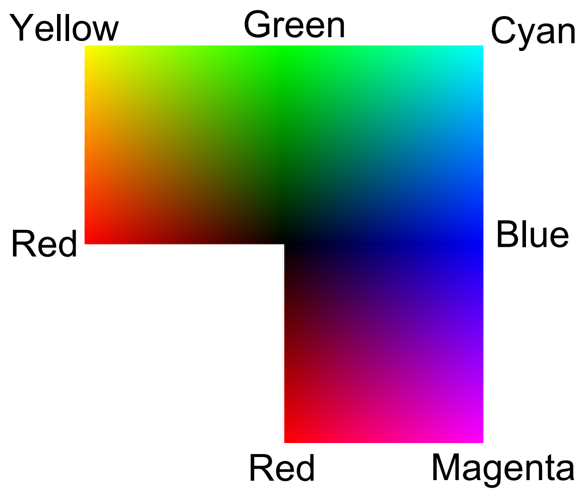

Figure 8: Flatten Schematic of the RGB Colour Cube

2.2.4 HSI Colour Model

HSI (Hue, Saturation, Intensity) also known as HSL (Lightness) colour space model is perception based colour space. Colour in the HSI colour space is defined by three components: hue, saturation, and intensity (or lightness). As Figure 9 shows, HSI colour space model is more human oriented colour space than RGB and CMYK colour space since human can pick a colour from colour wheel at the centre of intensity axis and make it brighter or darker by moves along with intensity axis.

A colour red is specified as a "reference colour" in HSI colour space. Because of that, $H = 0°$ and $H = 360°$ correspond to the colour red. For each component H,S and I, the component hue 'H' defines the dominant colour contained of the point. Saturation 'S' specifies the measurement of colour purity by amount of white added to the pure colour. Finally, intensity 'I' corresponds to the relative brightness of the point. Any point in the RGB colour space can also be specified using the HSI colour model and equation 4 illustrate define values of each components in HSI colour space from RGB colour space.

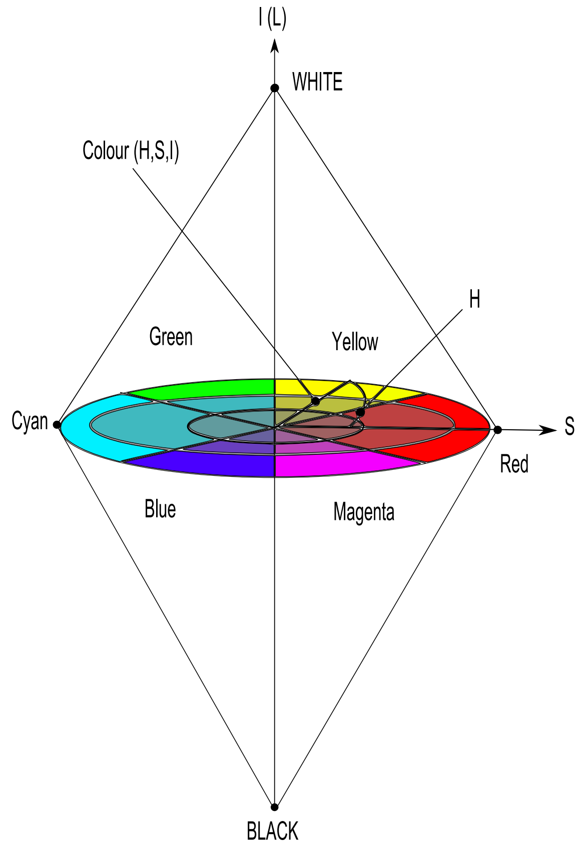

Figure 9: The HSI-Colour Space

$$H = \begin{pmatrix} \delta \\ 360^{\circ} - \delta \end{pmatrix} \begin{pmatrix} if B \leq G \\ if B > G \end{pmatrix}$$

$$\delta = \arccos(\frac{(R-G)+(R-B)}{\sqrt[2]{(R-G)^2+(R-B)(G-B)}})$$

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

$$I = \frac{R+G+B}{3} \tag{4}$$

2.3 Colour Image Formation

In digital colour image acquisition, most common way to acquiring a visible colour is using electronic image sensor which converts wavelengths of light into electric signal. Two most common image sensors are charge-coupled device (CCD) and complementary metaloxidesemiconductor (CMOS) active-pixel sensor.

Charge-Coupled Device (CCD)

A CCD is an analog shift register invented in 1970 at Bell Labs. A CCD enables the transportation of analog signals (electric charges) through successive stages (capacitors), controlled by a clock signal. [19] When light strikes a CCD's photo electric light sensor produces electrical charge that are later converted to voltage for digital reading. At this stage CCD shifts its charge into next CCD connected which results serialise analog signals to process into digital information. This collection of CCD is called CCD-chip and number of CCD consists of a CCD-chip relates to resolution of image to be acquired.

Complementary Metal Oxide Semiconductor (CMOS) Active-Pixel Sensor (APS)

An active pixel sensor (APS) is an image sensor consisting of an integrated circuit containing an array of pixel sensors, each pixel containing a photo-detector and an active amplifier. [20] Complementary Metal Oxide Semiconductor (CMOS) APS is most commonly used APS in digital imaging, produced by a CMOS semiconductor process.

The typical APS consist of photo-detector, a transfer gate, reset gate, selection gate and source-follower readout transistor. Figure 10 shows schematics of a three-transistor APS.

Like CCD-chip, an APS-chip consists of neighbouring APS. However, spatial gap

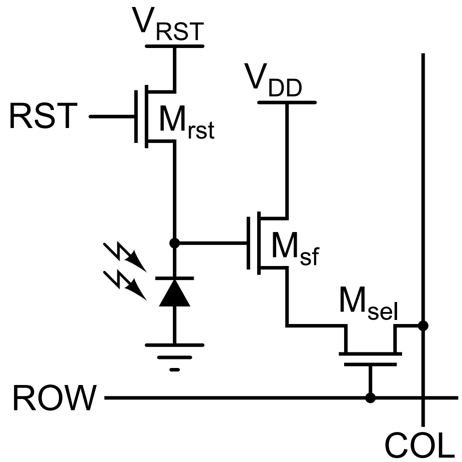

Figure 10: A Transistor-Level Schematic of A Three-Pixel, Photodiode-Based Active Pixel Sensor. *Diagram created by Gargan from Wikipedia*

between APS is usually larger than CCD which results darker acquired image than CCD-chip acquired. Beside of that the APS solves the speed and scalability issues of the passive-pixel sensor like CCD. The APS-chip can be accessed any spatial point, unlike CCD-chip which only accessible in serialisation. Furthermore APS consumes much less power than CCD and provides much faster response.

2.3.1 Colour Separation Mechanism

Since our visual system has three different cones which primarily respond to light in the red, green and blue parts of the spectrum, we need three types of sensors to measure the incident light. [14] Number of colour separation mechanism for CCD were introduced. 3CCD and Bayer colour filter array are most common.

3CCD Colour Separation Mechanism

3CCD mechanism is achieving colour separation is by having three separate CCD-chip designated to each colour separated by dichroic prism. This is simple and straight forward approach and loss of light information kept minimum. However, a single image requires three separate CCD-chip with same resolution of the image. Figure 11 shows 3CCD colour separation mechanism.

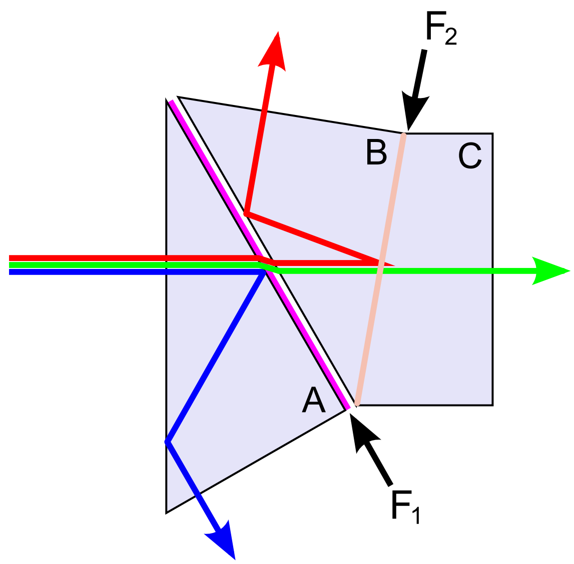

Figure 11: A Philips Type Trichroic Beam Splitter Prism Schematic, With a Different Colour Separation Order Than the Assembly Shown in the Photo. The Red Beam Undergoes Total Internal Reflection at the Air Gap, While the Other Reflections are Dichroic. *Diagram created by Gargan from Wikipedia*

Bayer Colour Filter Array

Colour filter array (CFA) is filter layer that arranging colour filters on square grid of image sensors. [21] Bayer colour filter mosaic was invented by Dr. Bryce E. Bayer of Eastman Kodak. A Bayer colour filter array consist of four filters in square shape consisting a red, two green, and a blue filter, because human eye could distinguish more green shades than red and blue shades. Since the red, green, and blue sensors are spatially separated, the image-sensors does not measure the incident light at a single point in space. [14] An interpolation algorithm is used to derived a colour pixel from these four component signals. Because of that, resolution of an image using Bayer filter less than number of image sensors used. Figure 12 and 13 shows Bayer filter used in image sensor arrary

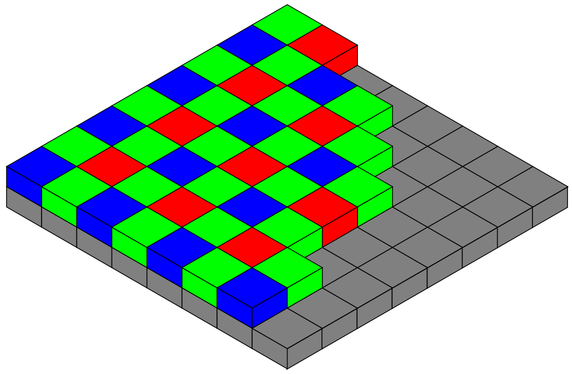

Figure 12: The Bayer Arrangement of Colour Filters on Image Sensor Array. *Diagram created by Cburnett from Wikipedia*

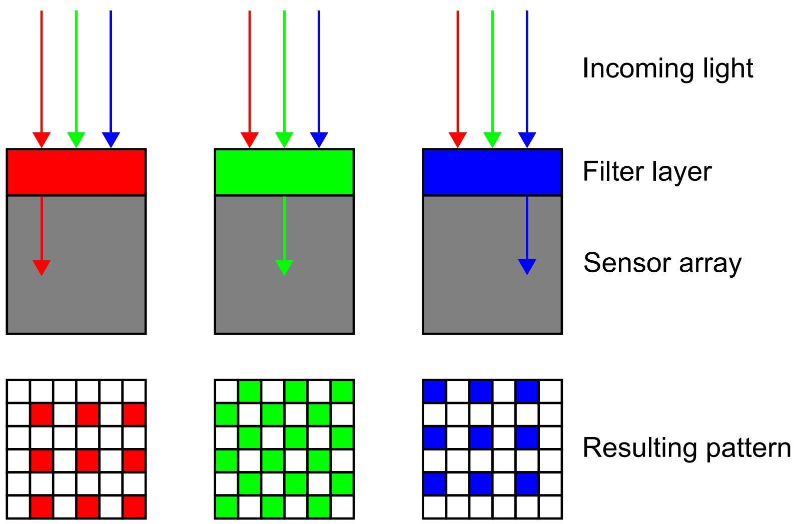

Figure 13: Profile/Cross-Section of Bayer Filter Layered Sensor. *Diagram created by Cburnett from Wikipedia*

2.4 Colour Representation in Binary

2.4.1 Colour Depth

A colour space model requires a certain form of binary representation in order to access and manipulate a colour in computers. The most common representation is called colour depth. Colour depth represents the total number of bits assigned to represent a colour in a given colour space. Commonly, colour depth is divided evenly for each colour space component and each component's normalised value is converted to fit in a given number of bits. Bits representation is usually in integer form and the fractional parts are either lost or rounded up or down. In RGB colour space, the common colour depth is 24-bits which consists of 8-bits of each component which represent 256 shades of red, green and blue that totals 16.7 million colours when combined. Each bit of colour depth increment yields double the precision of a component. In other words, each bit of colour depth decrement loses half its precision. Although higher colour depth represents more colours, it needs more memory and, demands higher computational efforts, and does not guarantee to yield better outcome.

2.5 Summary

We have briefly covered the nature of colour, the science of human colour vision, a number of colour space models, digital image acquisition, and colour representation. In order to reproduce colours, we mimic human colour vision, specifically cones and rod cells. Colour space models and colour acquisition methods are also oriented to follow human colour vision system to more correctly represent perceived colour. For example, ideas of having double amount of green filters on Bayer colour filter array and 16-bit colour depth which consists of 6-bits of green and 5-bits of red and blue colour component, shows how study of human colour vision influence digital colour representation. However, human colour perception is not concludes in retina, but in brain where colour constancy and optical illusion take place. Colour constancy is interesting human nature that we enables recognise a colour regardless of changes of light. It is also one of the most sought after field in computer vision study that would enables us to develop stable colour vision system.

Next chapter, we will review related literatures to study various techniques to mimic the colour constancy.

Chapter 3

Review of Related Literature

Colour constancy is desired human ability for computer vision researchers for correctly recognise colour under varying illumination changes. With colour constancy, human naturally matching perceiving colour from recognised colour in memory which helps greatly to recognise known objects. In computer vision research, colour classification ultimately aims recognise a colour from any illumination influence, and mimicking colour constancy is major step to achieve the goal. In this chapter, we are reviewing number of researches that tried to mimic colour constancy in various approaches.

3.1 Colour Segmentation

3.1.1 Indexing Via Color Histograms

Swain and Ballard, 1990 [15]

Swain and Ballard introduced a technique called histogram intersection for fast colour object segmentation and an algorithm called Histogram Back-projection for searching colour objects from crowded scenes.

An advantage of this technique is its orientation-invariant nature. The approach requires histogram intersection processing as well as the Histogram Back-projection algorithm applied on every single frame for real-time application.

Colour histogramming simplifies the identification of important features in the scene by counting the number of colours that occurred in the image array. Since histograms only have a finite number of colour pixels in the image array, histograms wouldn't change under translation and rotation about an axis perpendicular to the image plane. On the other hand, the change occurs only slowly under a change of viewing angle, change in scale and occlusion. Therefore a three dimensional object can be represented by a small number of histograms, corresponding to a set of canonical views.

Identifying objects via the histogram intersection technique requires prior training with multiple sample of the same object to extract common colour features (intersection in colour histogram). Equation 5 introduces the histogram intersection algorithm. Given a pair of histograms, I (image) and M (model), each containing n buckets, the fractional match value H is calculated. It's value falls within the range 0 and 1, where j ranges over each colour in the histograms.

$$H(I, M) = \frac{\sum_{j=1}^{n} \min(I_j, M_j}{\sum_{j=1}^{n} M_j} \tag{5}$$

This approach yields fast colour segmentation with high accuracy. However it is not scale invariant without some help from other parameters, such as distance.

Histogram Back-projection searches the location of the target object from the image array by making a convolution mask of colour histogram R which is the ratio of multidimensional colour histogram M divided by colour histogram of the image array I. The back-projected image is then convolved by a mask R that shows a peak location where the target object is expected.

The authors showed that the Histogram Back-projection algorithm can successfully identify multiple objects, with over 90% of accuracy. However, the authors does not clearly state whether the images were taken under constant illumination or not.

3.1.2 A Robust and Fast Color-Extracting using a Look up Table

Kim and Chung, 1999 [22]

Kim and Chung introduced a colour extraction technique using a look-up table. In this research, a scene is captured by a camera and stored into an image array. A look-up table is prepared for indexing colour values into classified colour indeces. Target colour regions are manually selected including surrounding edges, and then indexed using a look-up table. This look-up table is used for colour classification in real-time.

This research offered principal usage of a look-up table for colour classification, but no further processing is introduced. Lastly, no consideration of illumination changes nor automatic adjustment for colour classification is proposed.

3.1.3 Color recognition

Stachowicz and Lemke, 2000 [23]

Stachowicz and Lemke introduced a image identification technique using a simplified colour histogram.

A colour depth reduction technique is used for simplifying the 24-bit colour depth into the 3-bit colour using fixed threshold point. The simplified colour pixels in the image are used to construct the simplified colour histogram which has 8 predefined colour classes (i.e. red, green, blue, cyan, magenta, yellow, white and black). In the experiment, sample images were converted into the simplified colour histograms to the image identification. The identifications were partially successful when the sample image contains diversifying colour information that sufficient to generate unique simplified colour histogram.

3.1.4 Real-time, adaptive color-based robot vision

Browning and Veloso, 2005 [12]

Browning and Veloso introduced a colour object recognition technique using a smoothed YUV colour space histogram.

In this research, the experiments were conducted in an outdoor environment, under varying illumination conditions. The authors pointed out that a colour red is always more 'red', than say, any green surface and proved those claims using colour histogramming in the YUV colour space. An adaptive segmentation technique is employed followed by histogram building, then an application of thresholds to find objects. The authors introduced Algorithm 1 for image segmentation and thresholds adaptation. A pixel is tested against target colour likelihood mappings and

Algorithm 1: The Adaptive Colour-Based Robot Vision Algorithm [12].

Vision Algorithm(image)
  segmentImage(image):
    Result: Priori for each colour class $Pr(J)$
    foreach *pixel (p) in image* **do**
      foreach *colour class j* **do**
        if $P(p \in C_j) > \theta$ **then**
          $S = S \cup \{j\}$
        end
        $s = \max(S)$
      end
    end
  buildHistograms():
  adaptThresholds():
    foreach *class j* **do**
      $h' = conv(h, gauss_j)$
      $sp = stationaryPoints()$
      $dt = arg\max(sp.trough)$
      if $dt > minv$ **then**
        $t' = alpha * (dt - t) + t$
      end
    end
  findObjects():

classified with the highest priority colour class for which its likelihood is above that class's threshold. In order to avoid conflicting colours and lighting variations, the histogram is smoothed by convolution with a zero-mean Gaussian kernel operator. The target colours' likelihood mappings are defined by manually. This research proposed a colour classification technique using colour space clustering that is effective for an outdoor environment. The approach however, does not take into account using similar colours as targets (e.g. pink and orange, etc.).

3.1.5 A Fast Algorithm for Color Image Segmentation

Dong, Ogunbona, Li, Yu, Fan, and Zheng, 2006 [4]

Dong et al. introduced a colour image segmentation technique using a combination of K-means and a two-layer pyramid structure.

The K-means algorithm has been widely used for colour segmentation. However, the larger the image, the longer the computational time requirement. The two-layer pyramid structure cuts down the computational cost by constructing new images that are $\frac{1}{16}$th of the scale of the original images. These are used for the initial K-means operation for locating the objects. By acquiring the initial centres of the clusters from the $\frac{1}{16}$th scale, the actual location on the original image could be calculated. Figure 14 illustrates the two-layer pyramid structure for each colour component. A look-up table is used for referencing distance values between all the

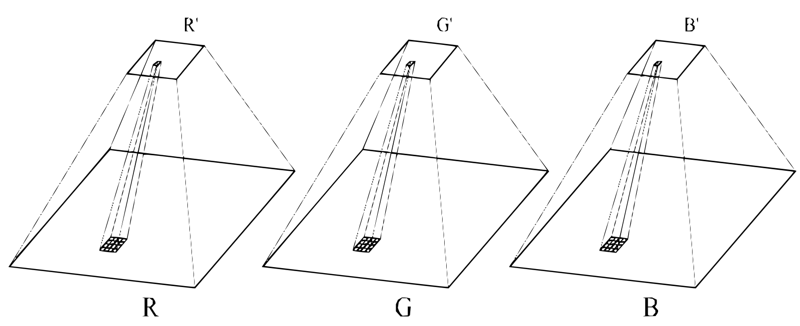

Figure 14: Two-Layer Pyramid Structure for Each Colour Component [4].

possible RGB component combinations. This research primarily demonstrated the feasibility of employing colour classification through the use of the K-means clustering algorithm.

3.1.6 Towards a calibration-free robot: The ACT algorithm for automatic online color training

Heinemann, Sehnke , Streichert , and Zell , 2007 [16]

Heinemann et al. proposed an algorithm for automatic on-line colour training (ACT) for the RoboCup [24] games.

The ACT algorithm employs a self-localisation technique. The initial input image is analysed to locate the current object position by mapping the perceived image into 2-D world coordinates. An omnidirectional input image is used for this technique and the ACT algorithm extracts two initial colour classifiers from the initial image; these are the colours Green and White. These two colours are easily extracted since green is the predominant colour in the field and white is assumed to be a colour with high intensity value. Equation 6 illustrates the filter function used for classifying the white colour, given the intensity of a pixel $I(p_{x,y})$ at position (x, y).

$$I(p_{x,y})\rangle \frac{1}{25} \sum_{i=x-2}^{x+2} \sum_{j=y-2}^{y+2} I(p_{i,j}) \tag{6}$$

The black colour is also extracted, as the robot chassis is coloured black. Black colours always appear in a fixed position in the input image. Further colour classifiers are extractable using a priori knowledge of the geometry of the field by segmenting known colour positions like the location of goals. The initial colour look-up table is trained from colour clusters resulting from segmentation the of the image. The colour cluster is established with a spherical shape, in a colour space which is centred at the mean value of a colour class with radius proportional to the standard deviation. Equation 7 describes the colour cluster definition for each colour class k, where the mean value is μ_k, and a standard deviation σ_k resulting from the colour values of previous cycles. The choice of η determines the responsiveness of the colour look-up table update.

$$c = (u, v, w) \in \{0, C_{\max}\}^3$$

$$\mu_{k,0} = \frac{1}{2}(C_{\max}, C_{\max}, C_{\max})$$

$$\sigma_{k,0} = \frac{\sqrt{3}}{2}(C_{\max})$$

$$\mu_{k,t} = \frac{1}{\eta + 1}\left(\eta\mu_{k,t-1} + \frac{1}{m}\sum_{i=1}^{m} c_i\right)$$

$$\sigma_{k,t} = \frac{1}{\eta + 1}\left(\eta\sigma_{k,t-1} + \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(c_i - \mu_{k,t-1})^2}\right) \tag{7}$$

The initial LUT is generated from the colour clusters. On the other hand, the colours of pixels that correspond to coordinates outside of the playable field are removed. To make the colour classifier adaptable to changes in illuminations, the ACT algorithm updates the LUT by accessing every 400th pixel, starting at a random pixel. The mean value of the corresponding colour class is compared to reposition the new mean value as well as the standard deviation. The thresholds used for adding and removing colour mappings from the LUT in different illumination conditions are also introduced.

The authors showed that the ACT algorithm is capable of correctly classifying colours quickly under changing illumination, both in static and moving situations. However, with strict requirements imposed by the algorithm, such as a priori knowledge of the geometry of the exploratory field, non-static objects such as a ball could not be classified without prior knowledge about its colour or position. These limiting factors raises some difficulty of applying the algorithm in general colour object recognition tasks.

3.1.7 Automatic On-Line Color Calibration using Class-Relative Color Spaces

Guerrero, Ruiz-del-Solar, Fredes, and Palma-Amestoy, 2008 [5]

Guerrero et al. introduced an automatic colour calibration technique using spatial relationship between colour classes.

Target colour classes are defined prior to on-line training, as spatial relationship between green and white colours. The on-line training process is divided into three stages, called class re-mapping, estimations combination, and LUT construction. Figure 15 shows a block diagram of the proposed colour classification system that trained and remapped colour classes. Estimations are combined to get a resulting estimation which is used to construct the LUT. Under the re-mapping classes

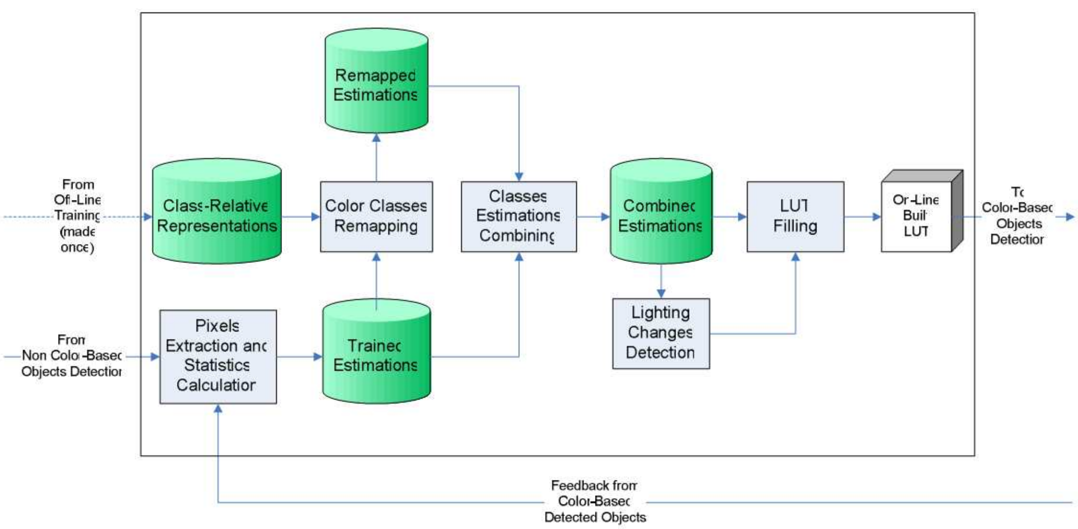

Figure 15: Block Diagram of the Colour Classification System [5].

stage, pixels are acquired through on-line training, and then labelled according to a colour class nearest in spatial distance. Each colour class has a limitation on number of member pixels to prevent over growth. On the other hand, in combining the estimations stage, colour classes are adjusted by shifting the spatial coordination including newly joined members. Lastly, in filling the LUT stage, the LUT is filled when any of colour classes are adjusted.

The authors showed that the result of the technique satisfies adapting to sudden illumination changes in a real-time environment. However, the authors did not provide solutions that solves the effects of spatial illumination changes or how to classify similar colours.

3.1.8 Adaptive recognition of color-coded objects in indoor and outdoor environments

Takahashi, Nowak, and Wisspeninter, 2008 [6]

Takahashi et al. introduced an adaptive colour recognition technique using probabilistic classification method with camera parameter adjustment in real-time.
A region segmentation was used for initial colour classification. An initial image captured from camera was segmented using Markov Random Field method and mean colour value of each segmentation was calculated. Colour classifiers are generated from the mean colour values using probabilistic classification method based on Mahalanobis distances, and threshold value was offered to reduce false positive colours. Equation 8 illustrates Mahalanobis distance between a colour value $x = (x_y, x_u, x_v)$ and distribution with mean $\mu = (\mu_y, \mu_u, \mu_v)$.

$$D_M(x) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)} \tag{8}$$

Figure 16 shows target environments captured by hyperbolic mirror attached camera with white and red paper ring fixed around the camera lens to provide reference colour point. Camera parameters such as Gain and Iris were controlled by PID controller to adapt illumination changes as keep colour classification valid by filtering input image. The authors tested the technique both indoor and outdoor, result from test showed that colour classification was stable in indoor. Figure 17, 18 shows YUV colour distribution of target environment, indoor and outdoor. Colour segmentation in colour space was significantly drifted in outdoor environment however, colour distributions do not overlap hence colour classification was still possible. This

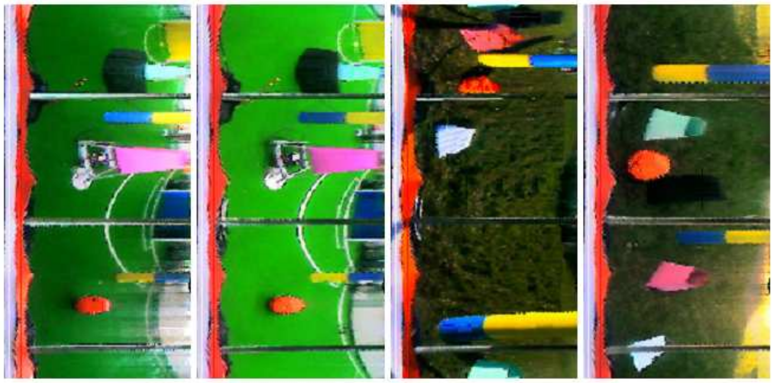

Figure 16: Captured Panoramic Images with PID Controller under Four Light Conditions [6].

research showed adjusting camera parameters to compensate colour shifting effects. However, this technique dependent to characteristics of the camera.

3.1.9 Mean-shift-based color tracking in illuminance change

Hayashi and Fujiyoshi, 2008 [7]

Hayashi and Fujiyoshi proposed a modified mean-shift algorithm for tracking 2-D colour blobs in an image.

The authors derived a colour-illuminance model that shows changes in RGB colour values under different illuminance settings with fixed white balance. The proposed research claims that conversion between an RGB colour value obtained from one iris parameter (F-number) to other RGB colour values obtained from another different iris parameter is possible. However, the proposed system is using an illuminance meter attached to the object to estimate the luminance level. Equation 9 shows the relationship between intensity I and F-number where f is the focal length and D is the iris diameter.

$$I \propto \left(\frac{D}{f}\right)^2 = \left(\frac{1}{F}\right)^2 \tag{9}$$

Figure 19 illustrates adjusting RGB colour value by F-number, RGB colour values at 1400 lx with $F = 5.6$ will be same as the RGB colour values at 700 lx with $F = 4$. A modified mean-shift algorithm is used for tracking a moving target colour

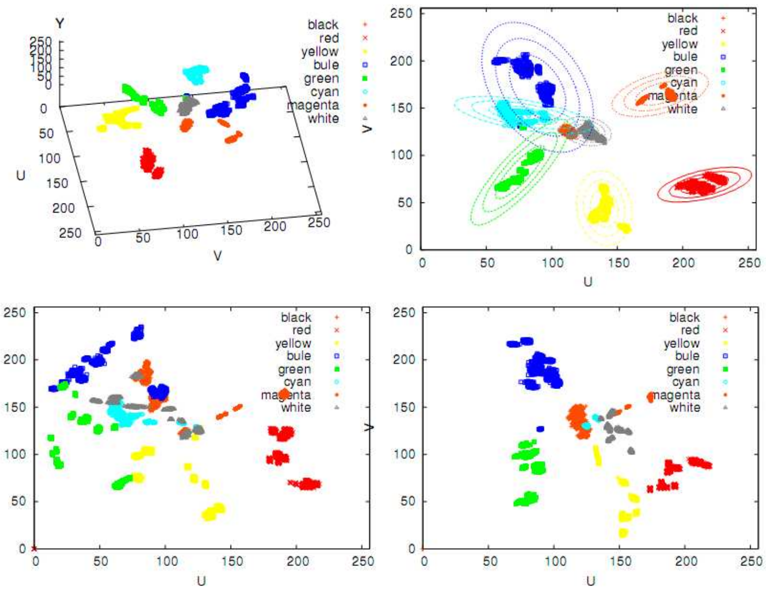

Figure 17: Plots of YUV Colour Distribution Indoors [6].

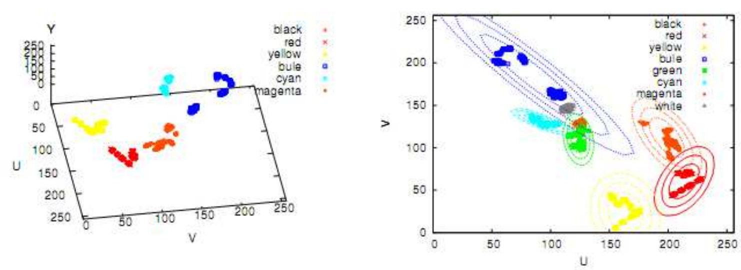

Figure 18: Plots of YUV Colour Distribution in Outdoor Environment [6].

blob under changing illumination conditions. An initial input image is analysed to estimate the current illuminance intensity of the environment. An estimate of the RGB values is then calculated based on the reference illuminance intensity to allow for more accurate colour classification. For object tracking, in the image array, a location weight map is built by calculating the distance travelled by the blob between successive frames. Each pixel is represented as a vector whose magnitude is based on its distance from a target colour model. Subsequently, a collection of pixel vectors is then used to calculate the spatial mean-shift vector that describes the moving target colour blob.

A new illumination intensity was found at the new position of target colour blob

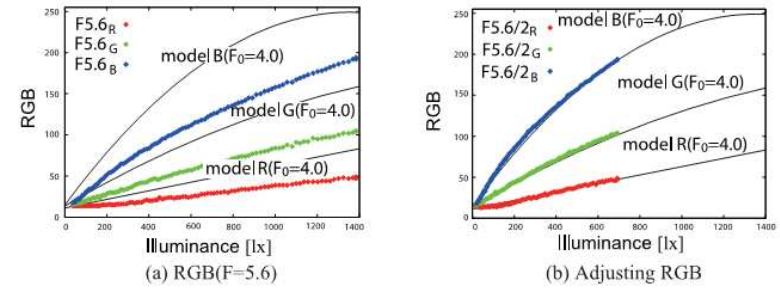

Figure 19: Adjusting RGB Colour Value by F-number [7].

by testing colour similarity at similar illuminance level around a local window of pixels.

The authors tested experiment of both single and multiple colour tracking under varying illumination ranging from 50 to 120 lx. The experiment results showed that even in rapid light change, the tracking adapts illumination well. However, the colour-illumination model with matching iris adjustment value requires provided for each different camera configuration.

3.1.10 Robust color classification using fuzzy rule-based Particle Swarm Optimization

Kashanipour A, Milani NS , Kashanipour AR , and Eghrary , 2008 [8]

Kashanipour et al. presented fuzzy rule-based colour classification method.

The fuzzy qualifiers were built based on HSL (HSI) colour space with three parameters; Hue, Saturation and Luminance as figure 20 shows HSL colour sphere with grids. Hue was divided into nine degree of colour as shown in figure 21, Saturation and Luminance were divided into three sub-intervals: low value, average value and strong value as illustrated in figure 22. Colour classifiers were defined with collection of fuzzy rules and Particle Swarm Optimisation (PSO) was used to optimise colour classification rules.

The experiment tested 12,000 test samples and result was showed best efficiency

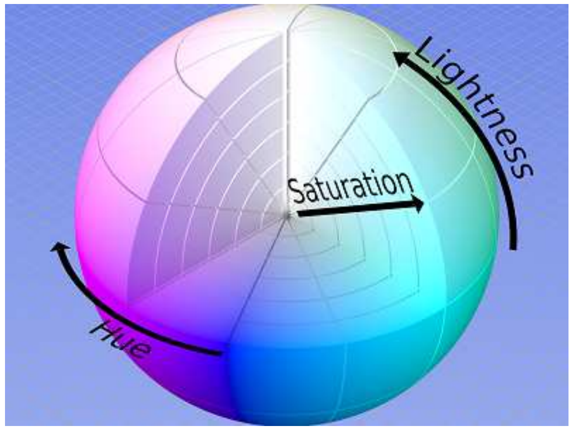

Figure 20: The HSL color Space Mapped to a Sphere, with Corner Cut-Away Shown. *Figure created by SharkD from Wikipedia*

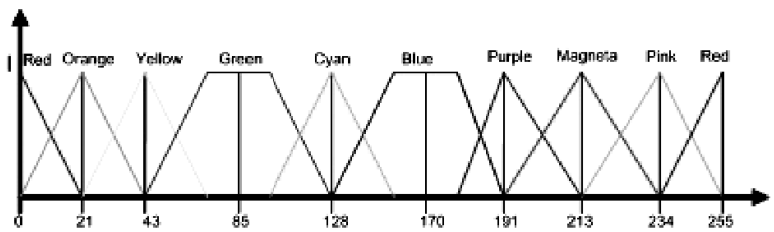

Figure 21: The Demension H [8].

rate of 75.43 % with 28 fuzzy rules for 10 different target colours. This research introduced using PSO technique to optimise fuzzy colour classification rules. However, number of samples used to generate optimised rules are limits its application.

3.2 Fuzzy Logic

3.2.1 Knowledge-Based Fuzzy Color Processing

Hildebrand and Fathi, 2004 [9]

Hilderbrand and Fathi introduced fuzzy logic technique to quality testing on resistance spot welding. This nondestructive quality testing method analyse digital image of welding spot. Figure 23 shows regions of interest of a resistance spot welding joint. The colour space used is called HS-colour space, which is derived from the HSI

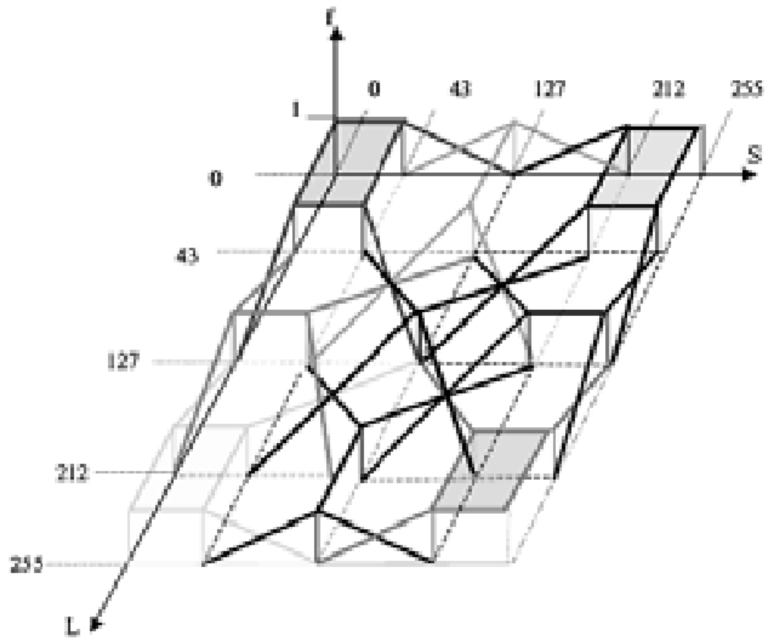

Figure 22: Dimensions L and S [8].

[h]

Table 1: Quality Criteria For Good And Poor Welding Spots. *From Knowledge-Based Fuzzy Color Processing, 2004*

Quality Criteria	Good Quality	Poor Quality
size of inner spot	8 mm	7.5 mm
size of outer spot	13.8 mm	9.9 mm
size of impact zone	18.7 mm	10.7 mm
shape of inner spot	circular	elliptical
shape of outer spot	circular	elliptical
shape of impact zone	circular	circular
colour of inner spot	blue - dark blue	light grey - light red
colour of outer spot	light blue	blue - red
colour of impact zone	red - dark red	red - light red

colour space, with the I-value set to 1.0. A colour classifier consists of eight cylindrical coordinates from the HS-colour space points: $S_{ia}, S_{ib}, S_{oa}, S_{ob}, C_{ia}, C_{ib}, C_{oa}$, and C_{ob}, where each point represents one corner of the fuzzy set, the letters S and C represents the terms support and core. Moreover, the letters i and o corresponds to the terms inner and outer. Figure 24 shows a cross-section of the HS-colour space with fuzzy set definitions. Table 1 illustrate use of linguistic terms to design fuzzy membership functions. The use of fuzzy logic enable to use linguistic terms rather than by names for wavelength or wavelength intervals to describe welding result.

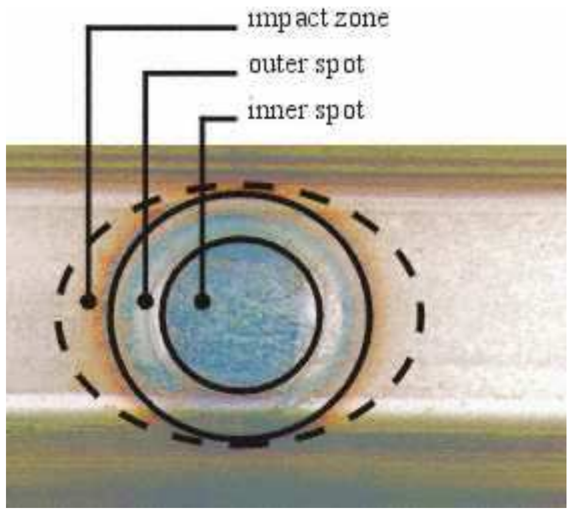

Figure 23: Regions of Interest of a Resistance Spot Welding Joint [9].

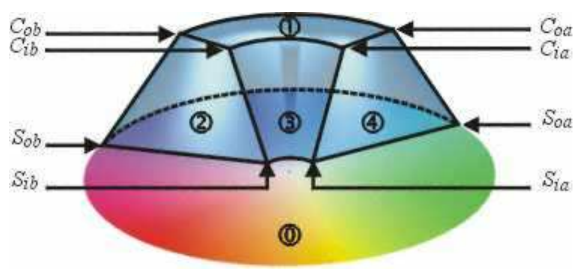

Figure 24: Fuzzy Set, Defined Over the HS-Colour Space [9].

This technique can be used as naming colours by using fuzzy logic to classify a colour from various colour spaces.

3.3 Fuzzy Colour Contrast Fusion (FCCF)

3.3.1 Dynamic Colour Object Recognition Using Fuzzy Logic

Reyes and Dadios, 2004 [10]

Reyes and Dadios introduced new colour classification technique called Reyes-Dadios Colour Contrast Fusion (RDCCF or FCCF) using fuzzy logic to contrast colour pixels on modified rg-chromaticity colour space.

An object in plain solid colour patch reflects a multitude of colours and these colours tend to form a cluster when projected in colour space. This cluster of colours are unstable even when object being tracked down under a fixed illumination because quantum electrical effects in the camera sensor chip easily distorts the colour captured and cannot be prevented.

Reyes and Dadios introduced Fuzzy colour constancy algorithm with colour classification scheme to stabiles colour cluster for colour recognition. Figure 25 illustrate overall colour classification algorithm they introduced.

In order to recognise the specific colour from the pixel, FCCF requires predefined

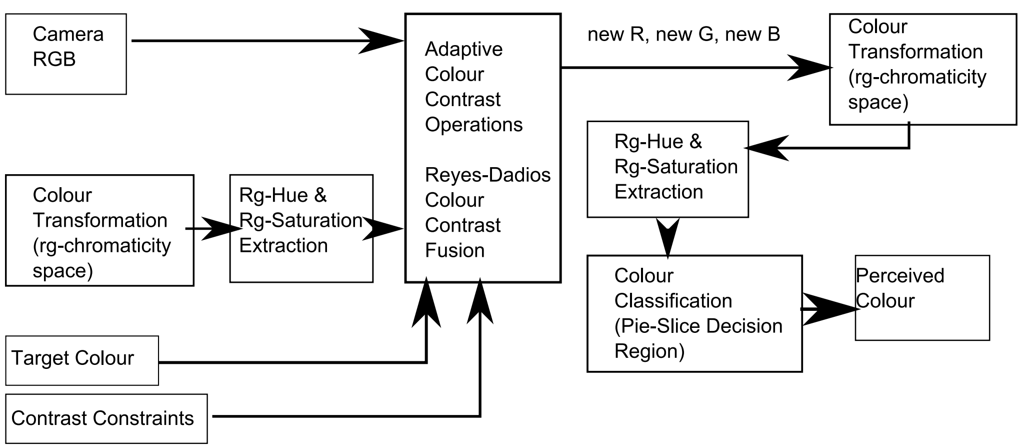

Figure 25: Colour Contrast and Classification System Architecture [10].

set of parameters such as contrast angle, classification angle and colour contrast operations which found by manually. The captured colour pixels from camera converts to rg-Hue and rg-Saturation colour components in modified rg-chromaticity colour space. Equation 10 shows conversion formulae

$$r = R/(R + G + B)$$

$$g = G/(R + G + B)$$

$$rg - Saturation = \sqrt{(r - 0.333)^2 + (g - 0.333)^2}$$

$$rg - Hue = \tan^{-1} \frac{(g - 0.333)}{(r - 0.333)} \tag{10}$$

rg-Hue and rg-Saturation are used as angle and radius to point in a modified rg-chromaticity colour space where origin moved from (0,0) to (0.333, 0.333) which points to the white point. Figure 26 shows rg-chromaticity colour space and origin shift position. A pie-slice colour decision region is used for whether colour contrast

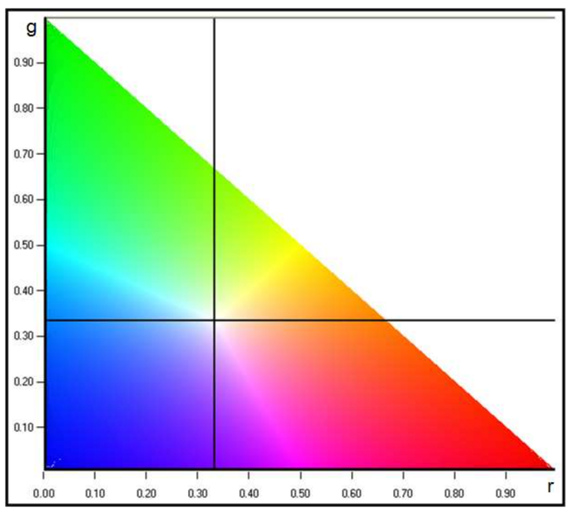

Figure 26: rg-Chromaticity Colour Space with Origin Shift Position [10].

operation is required. When a colour point is within pie-slice contrast angle of colour decision region, the point's original RGB value applied contrast operations to compensate for hue and saturation drifting in a colour space. Figure 27 shows pie-slice colour decision region.

Two complementary colour contrast operators (Enhance and Degrade) were intro-

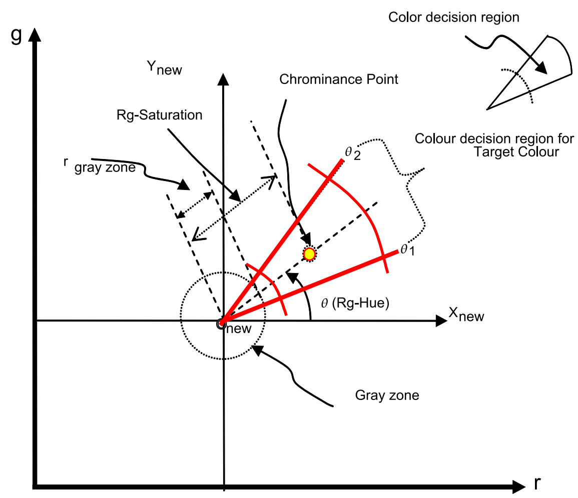

Figure 27: Pie-Slice Colour Decision Region in Modified rg-Chromaticity Colour Space [10].

duced that employed Sigmoid (Logistic) and Logit functions. In contrast enhance operation, any signal greater than or equal to the threshold will be amplified, while signals less than the threshold will be attenuated. In contrast degrade operation, pulls any given signal closer towards the threshold setting. Each colour component assigned colour operator independently at different degrees of level with threshold equal to 0.5. Equation 11 and 12 shows sigmoid and logistic functions.

$$\alpha = \begin{cases} 2{\mu_\alpha}^2(y) & 0 \leq \mu_\alpha(y) < 0.5 \\ 1 - 2[1-\mu_\alpha(y)]^2 & 0.5 \leq \mu_\alpha(y) \leq 1 \end{cases} \tag{11}$$

$$\alpha = \begin{cases} 0.5 - 2(1 - [\mu_\alpha(y) + 0.5]^2) & 0 \leq \mu_\alpha(y) < 0.5 \\ 0.5 + 2[\mu_\alpha(y) - 0.5]^2 & 0.5 \leq \mu_\alpha(y) \leq 1 \end{cases} \tag{12}$$

Successive application of contrast enhancement or degradation causes its effect more rapidly as shown figure 28 and 29. The RGB value applied contrast operations are called *new* R, *new* G, and *new* B that converted to new rg-Hue and new rg-Saturation to see whether these new point in the modified rg-chromaticity colour space is within colour classification angle of pie-slice decision region which classified as target colour.

A test suite for measuring accuracy of colour classification is derived as form of true positive (TP), false positive (FP), true negative (TN) and, false negative (FN).

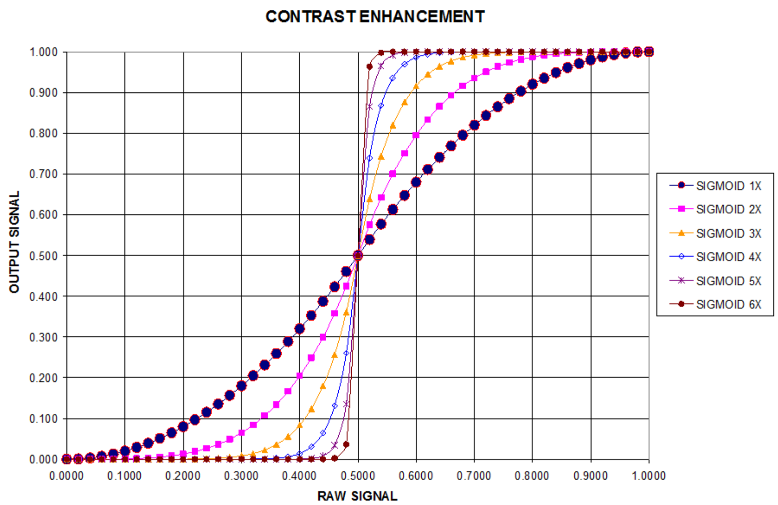

Figure 28: Colour Contrast Enhance Operator (Sigmoid / Logistic Function) [10].

- When *new* R, *new* G, and *new* B classified as target colour and it is within boundary of real target colour, it is true positive.
- When *new* R, *new* G, and *new* B classified as target colour and it is not within boundary of real target colour, it is false positive.
- When *new* R, *new* G, and *new* B not classified as target colour and it is not within boundary of real target colour, it is true negative.
- When *new* R, *new* G, and *new* B not classified as target colour and it is within boundary of real target colour, it is false negative.

Figure 30 shows original test image with pink colour patches. Figure 31 shows true positive and false positive against target colour Pink. Using FCCF under real-time environment requires look-up table (LUT) building process prior to used in real-time environment, because re-calculating *new* R, *new* G, and *new* B for every pixel in every frame is not practical. For every possible combination of RGB values, *new* R, *new* G, and *new* B are calculated and classified colour name stored into LUT against corresponding RGB value. In real-time environment, a perceived RGB value simply referenced to LUT to find which target colour the pixel is belong to. Algorithm 2 describes LUT building process. FCCF was tested on image as shown in figure 30, six different colour patches under varying illumination and

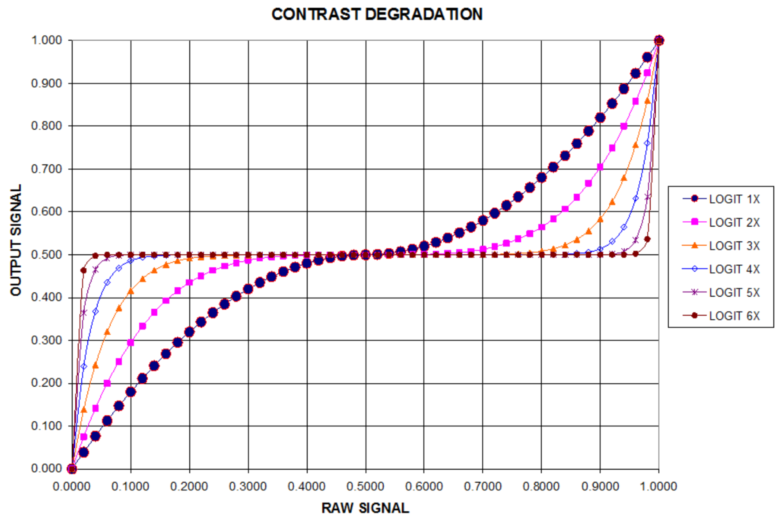

Figure 29: Colour Contrast Degrade Operator (Logit Function) [10].

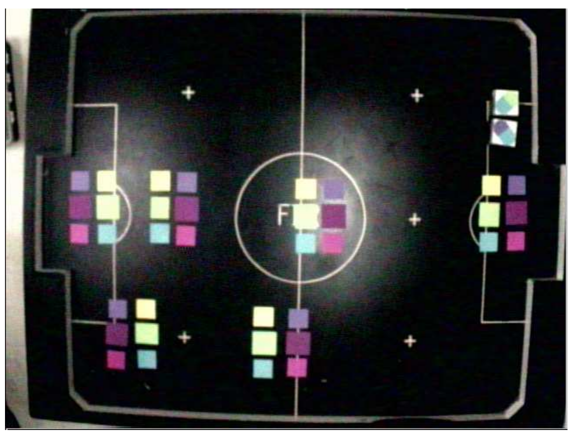

Figure 30: Test Image with Pink Colour Patches in the Middle [10].

correctly classified all six colours and objects. This research opens new perspective of colour classification that employs fuzzy logic and addressed classifying similar colours. However, colour classifiers were required to manually calibrated.

3.3.2 Identifying Colour Objects with Fuzzy Colour Contrast Fusion

Reyes and Messom, 2005 [1]

Reyes and Messom continued research of FCCF and applied FCCF on various colour

Figure 31: Colour Classified Image. True Positive Pixels are in Light Blue, False Positive Pixels are in Blue Colours.

Algorithm 2: Look-Up Table Building Algorithm

for *Every possible Red values* **do**
  for *Every possible Green values* **do**
    for *Every possible Blue values* **do**
      if *colour value is classified as colour* t **then**
        $L = (R \ll (depth*2)) + (G \ll depth) + B$;
        $LUT[L] = t$;

depth is the colour depth values of each colour channel in source image.

spaces.

YUV and HSI colour spaces were selected for test efficiency of FCCF since YUV and HSI colour spaces are the most popular colour spaces in colour classification research: YUV colour space is standard image format for most cameras, and HSI colour space is often used because of selectiveness of Hue component. In order to apply contrast angle and radius on 2-D plain in YUV and HSI colour spaces, intensity part (Y and I) was dropped from the colour spaces. Table 2 shows colour classification angles and radii in each colour space for three target colours: Light Blue, Blue, and Violet. Table 3 shows colour contrast rules for three target colours. Figure 32 and table 4 shows results of applying colour contrast fusion in three colour spaces. This research offered that FCCF is effective in various colour space without modification.

Table 2: Colour Descriptors for the Target Colours [1].

	Angle		Radius		Contrast Constraints	
rg SPACE	Min	Max	Min	Max	Min	Max
LightBlue	149.688	152.604	0.0393	1	147.996	153.108
Blue	152.604	166.284	0.1326	0.4175	149.292	174.312
Violet	211.536	262.152	0.0765	0.1168	210.636	264.024
YUV SPACE						
LightBlue	283.284	289.044	10.476	55.71	276.912	296.496
Blue	277.164	317.088	45.288	136.422	269.712	324.936
Violet	327.384	360	35.676	75.15	319.032	338.688
HSI SPACE						
LightBlue	0	0.001	0.0589	0.2434	0	1
Blue	0	205.056	0.2621	0.9418	0	209.628
Violet	221.148	259.02	0.112	0.2763	220.68	260.856

Table 3: Colour Contrast Rules for Each rg-chromaticity, YUV and, HSI Colour Spaces [1].

	Contrast Operation			R	G	B
rg SPACE	R	G	B	Level	Level	Level
Light Blue	Degrade	Enhance	Degrade	1	1	1
Blue	Degrade	Enhance	Degrade	1	1	1
Violet	Degrade	Degrade	Degrade	1	2	1
YUV SPACE						
Light Blue	Degrade	Enhance	Degrade	1	1	1
Blue	X	Enhance	Degrade	0	2	1
Violet	Enhance	X	X	3	0	0
HSI SPACE						
Light Blue	X	Enhance	Enhance	0	2	1
Blue	X	Enhance	Enhance	0	2	1
Violet	Degrade	Degrade	Degrade	1	2	2

Table 4: False positive and true positive rates for the Colour Contrast Fusion Algorithm in rg-Chromaticity, YUV, and HSI Colour Spaces [1].

	Light Blue		**Blue**		**Violet**	
Colour Space	**FP**	**TP**	**FP**	**TP**	**FP**	**TP**
rg	0.0037	0.5908	0.0226	0.7139	0.0247	0.7786
rg+	0.0004	0.6183	0.0003	0.6023	0.0004	0.5957
YUV	0.0031	0.7324	0.0015	0.7844	0.0011	0.7907
YUV+	0.0008	0.6473	0.0005	0.6821	0.0007	0.7447
HSI	0.0077	0.6149	0.1793	0.7984	0.0355	0.8288
HSI+	0.0005	0.6183	0.0105	0.7733	0.0004	0.6746
Label with '+' sign indicate utilization of colour contrast fusion.						

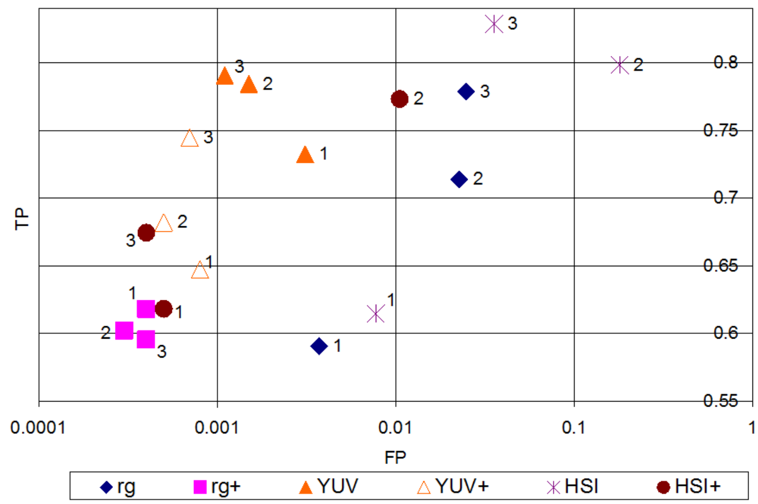

Figure 32: Results of Applying Colour Contrast Fusion in rg-Chromaticity, YUV, and HSI Colour Spaces [1].

3.3.3 Hybrid Fuzzy Colour Processing and Learning

Playne and Reyes, 2008 [11]

Playne and Reyes continued the research of FCCF and proposed Motion-based Predictive Colour Learning algorithm (MPCL) and Colour Contrast Rule Extraction algorithm (CCRE) for automatic colour classification and optimisation.

FCCF provided highly accurate colour classification results however, requires manually calibrated colour classifiers which makes impractical for rapid deployment. To overcome the barrier, Playne and Reyes developed the MPCL algorithm that generates colour classifier values while tracking circular shape target colour object in real-time. Figure 33 illustrates the MPCL algorithm. When a target coloured object is specified from the real-time scene, the algorithm generates circle that includes target colour pixels by calculating height and width of the object as shown in figure 34 and equation 13.

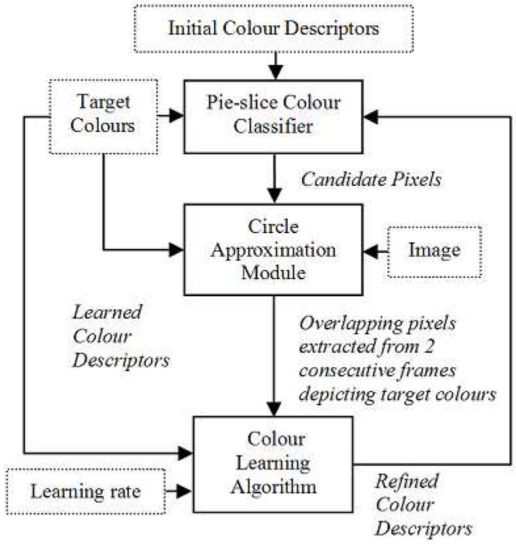

Figure 33: The MPCL Algorithm [11].

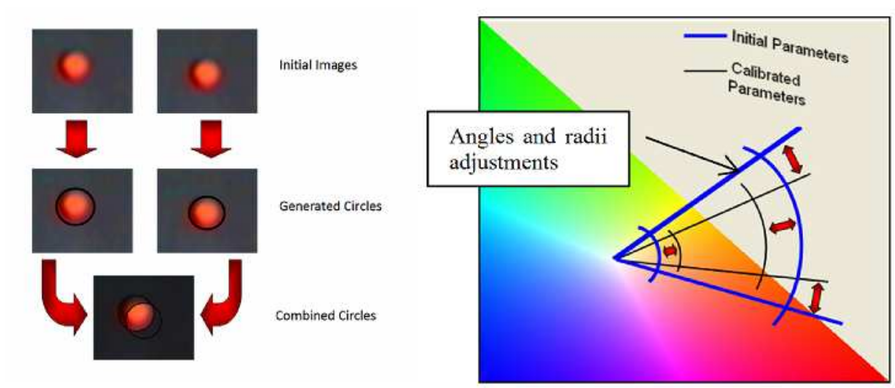

Figure 34: Extracted Object Colour Pixels From Two Consecutive Frames and Corresponding Colour Classification Variable Changes [11].

$$
\begin{aligned}
x_{centre} &= \sum_{i=0}^{n} xi \\
y_{centre} &= \sum_{i=0}^{n} yi \\
height &= \max(x_{centre}, y) \\
width &= \max(x, y_{centre}) \\
radius &= \frac{height + width}{4}
\end{aligned}
\tag{13}
$$

After the target circle is discovered, the algorithm generates moving average of the maximum and minimum rg-Hue and rg-Saturation values as shown in equation 14.

$$rgHue_{max} = \frac{rgHue_{max}(i-1) + \max(rgHue)}{i}$$
$$rgHue_{min} = \frac{rgHue_{min}(i-1) + \min(rgHue)}{i}$$
$$rgSaturation_{max} = \frac{rgSaturation_{max}(i-1) + \max(rgSaturation)}{i}$$
$$rgSaturation_{min} = \frac{rgSaturation_{min}(i-1) + \min(rgSaturation)}{i} \quad (14)$$

Once colour classification variables for angles and radii are found from the MPCL algorithm, the CCRE algorithm refine the colour classification accuracy by finding optimised colour contrast rule. The CCRE algorithm finds optimised colour contrast rule by iterate every possible colour contrast rules to calculate colour accuracy score which leads to discover best colour contrast rule as algorithm 3 describes calculating colour classification accuracy score. This research allowed defining

Algorithm 3: *CCRE(image, targetbounds),* Scoring Formula [11].

1. For each target object calculate an individual score: $score_i = \frac{hits_i}{area_i}$

   if $hits_i < \frac{1}{n}\ area_i$ then $score_i = 0$; where $n = 4$ (empirically found)

2. Calculate average score:

   $avescore = \frac{\sum_{i=1}^{ntargets} score_i}{ntargets}$; where: $ntargets$ is the number of targets.

3. Calculate a general score:

   $genscore = \frac{Totalhits}{Totalhits + Totalmisses}$

4. Final score:

   $finalscore = (0.6\, avescore) + (0.4\, genscore)$

5. Adjust score to account for misclassifications:

   $if(Totalhits > 0)$

   $finalscore = finalscore - (\frac{Totalmisses}{Totalhits})$

colour classification values rapidly. However, does not provide adequate solution when multiple similar colour objects required to classified as the MPCL algorithm does not differentiate colour classification angle and colour contrast angle.

3.4 Summary

We have reviewed related literatures in the fields of colour extraction, segmentation, classification and object recognition from 1990 onwards. Prior to 1990, colour object recognition was not a very viable task due to the prevailing hardware limitations during those times: slow CPU and small memory capacity, and the unavailability of fast digital image acquisition systems. As a consequence, object recognition researches were mostly limited in the gray scale level. From 1990, the number of researches in the field of colour object recognition has increased exponentially as the computer processing power leaped almost every year.

Many techniques for object recognition has taken the colour-based approach. Of these approaches, colour histograms emerged to be popular because it is simple to construct, fast, and it is view and shape invariant. Swain and Ballard used colour histograms to extract features in the scene with controlled lighting conditions in 1990 [15]. One of the promising techniques proposed in this work is the Variable Colour Depth colour representation, along with the VCD LUT. So far, to the best of our knowledge, there is only one existing algorithm by Stachowicz and Lemke, that adheres to the same idea of using colour depth reduction for improved colour discriminability [23]. They proposed an image identification technique using a simplified colour histogram in 2000 [23]. They also introduced a colour depth reduction technique that is used to construct a simplified colour histogram. However, their proposed approach is too simplistic, using only 3-bits for colour pixel classification. The algorithm fails for colour object recognition of objects lacking colour diversity. In addition, the presence of similar colours were not investigated as well. In a similar fashion, Browning and Veloso in 2005 also used colour histograms for classifying colour objects, but in an outdoor environment with promising results. Their approach employs an adaptive thresholding technique, and claims that the algorithm works even when there are changes in the illumination. Nevertheless, the presence of similar colours were never taken into account [12]. To speed-up the process of real-time colour classification, Kim and Chung used a look-up table approach in 1999 [22]. However, this entails a burden of calibrating the classifier manually. Moreover, simplification techniques through scaling were also introduced to speed-up the colour segmentation task. Dong et al. showed fast image segmentation using the K-means clustering algorithm with a layered pyramidal structure in 2006 [4]. The proposed technique worked fast even for very large images because their algorithms approximates the centre of the colour cluster from a $\frac{1}{16}$th scaled version of the original image. On the other hand, techniques based prior knowledge of the scene's geometry were employed to aid the colour training task. Heinemann et al. in 2007 [16] proposed a colour training algorithm for the RoboCup [25], four-legged (AIBO) league that matches an acquired colour information with prior knowledge of the geometry of the playing field. More recently, knowledge of the spatial relationship between colour classes in a colour space was utilised to develop an adaptive colour classification technique that works even when there are sudden illumination changes [5]. However, spatial illumination variations in the field and discrimination between similar colours were not taken into account. In addition, hardware-assisted adaptive illumination invariant techniques were introduced. Takahashi et al. in 2008 [6] employed a mechanical PID control to automatically adjust camera parameters such as the iris and the gain to adapt to illumination changes in the target environment. A reference red ring around the lens was used to determine when and how much adjustments for the iris and gain parameters need to be performed. Also, Hayashi and Fujiyoshi in 2008 [7] derived a colour-illuminance model that shows changes in RGB colour values under different illuminance settings with fixed white balance. The proposed research claims that conversion between an RGB colour value obtained from one iris parameter (F-number) to other RGB colour values corresponding to a different iris parameter is possible. However, the proposed system is dependent on an illuminance meter attached to the object to estimate the luminance level. Lastly, fuzzy colour processing algorithms are employed rather sparsely. Many of the existing techniques employ fuzzifications of the colour classes to solve ambiguity issues. On the other hand, in this research, the fuzzy techniques are mainly employed for colour corrections, to compensate for the illumination effects. To mention some related works on fuzzy approaches, Kashanipour et al. proposed a colour classification technique using fuzzy rule-sets operating in the HSI colour space and optimised with particle swarm optimisation technique in 2008 [8].

In another research, Hilderbrand and Fathi in 2004 [9] analysed welding spots through colour inspection and shape estimations. The proposed research employed fuzzy logic that enabled the use of linguistic terms rather than numeric values to describe the quality of welding spots. The colour space used is called HSI-colour space, with the I-value set to a constant value of 1.0. A colour classifier consisting of eight cylindrical coordinates from the HS-colour space represents the parameters of the fuzzy sets.

Since Reyes and Dadios introduced FCCF in 2004 that used pie-slice decision region and colour contrast rule to classify colour objects under varying illumination, continuous researches were followed. In 2005, Reyes and Messom applied FCCF on various colour spaces. [1] In 2006, Reyes et al. applied FCCF with Adaboost training for automatic colour classification. [26] Most recently Playne and Reyes introduced colour learning technique using FCCF in 2008 [11].

In the next chapter we will introduce Variable Colour Depth concept with supporting algorithms as well as Fuzzy-Genetic colour classifier search strategy.

Chapter 4

Central Thesis

4.1 Variable Colour Depth

When computers access and manipulate a colour explicitly, the colour space model representation must be in binary form. The most common defining characteristic is called colour depth. Colour depth represents the total number of bits assigned to represent a colour in a given colour space. Commonly, the colour depth is divided evenly for each colour space component and each component's normalised value is converted to fit in a given number of bits. Bits representation is usually in integer form and the fractional parts are either lost, or rounded up or down when converting from a higher to lower resolution and vice-versa. In RGB colour space, the common colour depth setting is 24-bits, which consists of 8-bits per colour component. This represents the 256 shades of red, green and blue that totals 16.7 million colours when combined. Each bit of colour depth increment yields double the precision of a component. The corollary is also true, each bit of colour depth decrement loses half its precision. Although higher colour depth represents more colours, it needs more memory and, demands higher computational efforts, and does not guarantee to yield better outcome.

Variable colour depth is a non-conventional approach to representing colour for digital image processing. In the RGB colour space, three components are usually represented using equal magnitudes (i.e. 0..255), and therefore, represented using equal number of bits. In the Variable Colour Depth representation, each colour component could be represented using varying number of bits. For example, a colour depth of 6-bits for red, 8-bits for green, and 8-bits for blue in the RGB colour space means that the red component have a quarter of resolution less than the green and blue components. However, it does not mean that the red component would only represent a quarter of the possible values covered by the other colour components but its colour gradient is only less smoother than other components.

Table 5 shows the bit representation of Variable Colour Depth and the corresponding values of the colour components. Figure 35 and 36 illustrates the normalised output values from different colour depths.

Table 5: Sample Variable Colour Depth Representations of the Normalised Colour Component Values 0.8 Red, 0.5 Green, and 1.0 Blue

Colour Depth				Binarized Representation			Nomarlised Value		
Total Bits	Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
24	8	8	8	11001100	01111111	11111111	0.8	0.498039	1
22	8	7	7	11001100	0111111	1111111	0.8	0.496063	1
22	8	6	8	11001100	011111	11111111	0.8	0.492063	1
21	5	8	8	11000	01111111	11111111	0.774194	0.498039	1
21	6	7	8	110010	0111111	11111111	0.793651	0.496063	1
21	7	7	7	1100101	0111111	1111111	0.795276	0.496063	1
18	7	6	5	1100101	011111	11111	0.795276	0.492063	1
18	6	6	6	110010	011111	111111	0.793651	0.492063	1
18	8	5	5	11001100	01111	11111	0.8	0.483871	1
18	6	6	6	110010	011111	111111	0.793651	0.492063	1
17	7	5	5	1100101	01111	11111	0.795276	0.483871	1
17	5	6	6	11000	011111	111111	0.774194	0.492063	1
15	5	5	5	11000	01111	11111	0.774194	0.483871	1

When a colour pixel is represented as a combination of colour components in the digital colour space, each component holds a value in some memory storage. In addition, each component reflects the component's level of contribution to the composition of the colour, as well as it's influence on the pixel's position in the colour space. The size of memory storage determines the number of possible levels between the minimum and the maximum that can be assigned to each component. We use the term 'colour component depth' to count the number of bits required to hold any

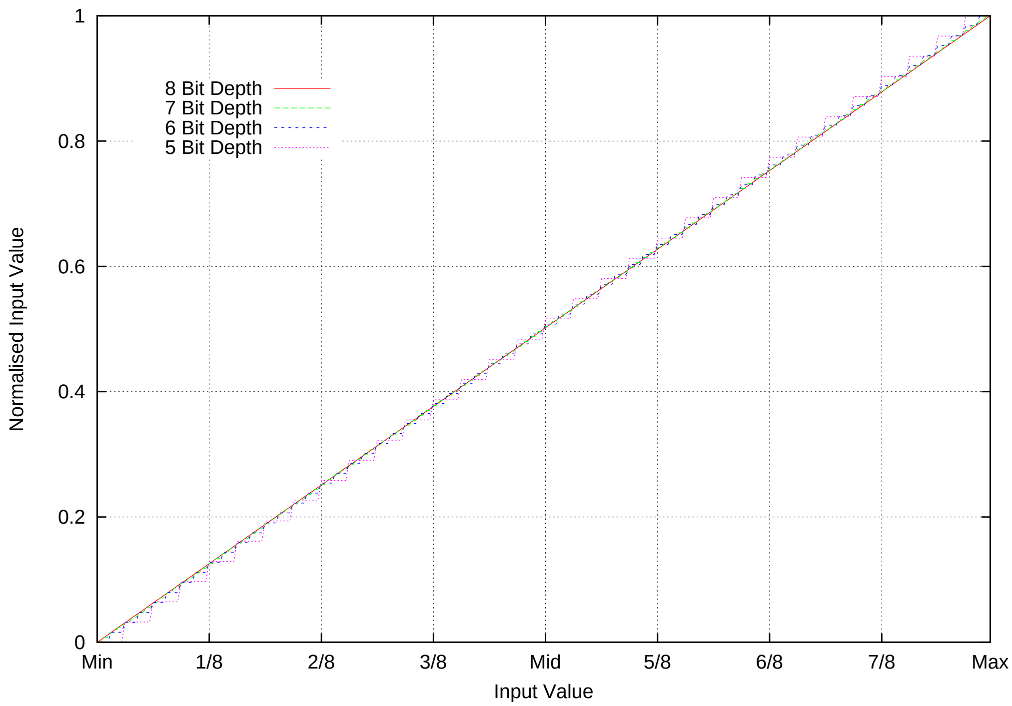

Figure 35: Normalised Input Values from Various Colour Depth

given component. On the other hand, and term 'colour depth' is used to express the total number of bits to represent all of the components of a single colour in any given colour space. When the colour depth changes, it is common to adjust each colour component depth uniformly in the colour space altogether. When the colour depth decreases, the colour information is simplified and memory requirement is reduced. The simplified colour information affects the intended or unintended results on the image quality, such as loss of colour shades or distinguishing artifacts. Figure 37 illustrates the effects of colour depth reduction. The details of the clouds clearly show the loss of colour shades as a result of colour depth reduction.

In general, the reduction of colour depth degrades the image quality. As a result, similar colours are merged into a single colour as shown in figure 37, from (d) the windows in the left hand side were merged together and they are no longer distinguishable from each other. However reducing the colour depth from 24-bit RGB colour depth to 15-bit RGB colour depth may not affect the human visual experience substantially. For example, figure 38 shows an image with the original

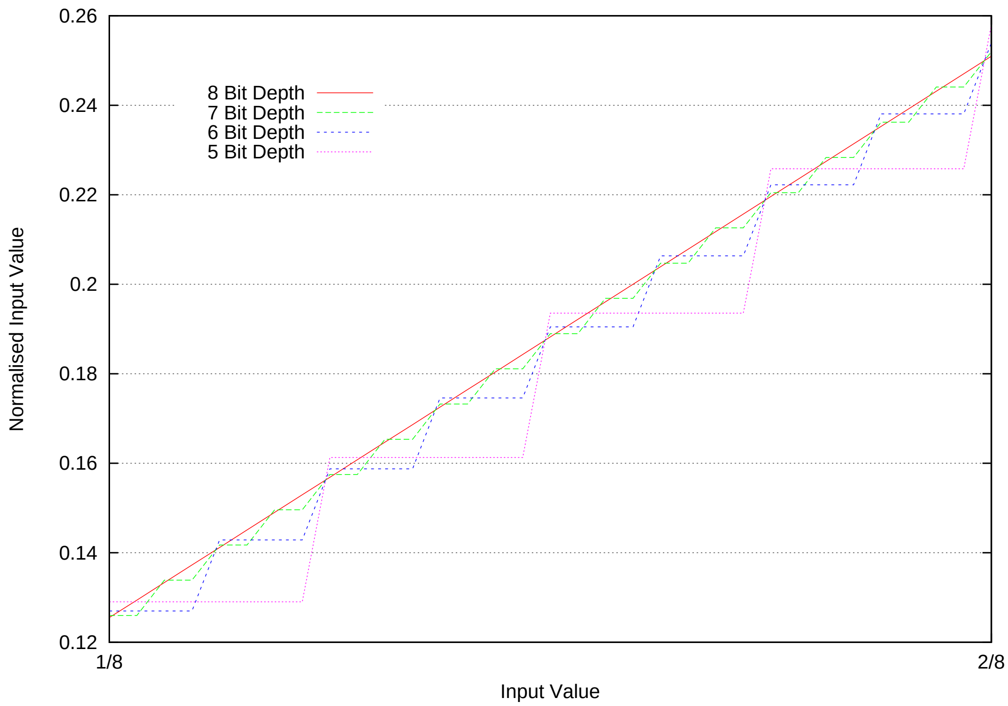

Figure 36: Enlarged Section of Normalised Input Values from Various Colour Depth Input Values from 1/8 to 2/8

24-bit RGB colour depth, and the reduced 16-bit and 15-bit RGB colour depths. Although the 24-bit RGB colour depth image used more than ten times the number of colours used in other images, the visual differences are hardly noticeable. It particulary worked well when the 24-bit RGB colour depth was converted to 16-bit RGB colour depth which consisted of 5-bits for red and blue, and 6-bits the for green component. The main reason behind this is that the human vision system is particulary more sensitive to medium visible wavelengths (yellow-green) than other visible wavelengths [19].

Colour information reduction has been applied by many researches already, and the main impetus generally is to get rid of colour information that will not compromise the quality of the image. YUV for instance usually preserves only 1 value of U and V for every 4 pixels in an image, and this is usually specified as 4-1-1 (sometimes 4-2-2 in other systems). In this research, we are not reducing the colour depth of the image for storage purposes, but we are reducing the colour depth of the image for analysing it; that is, for colour classification. To the best of our

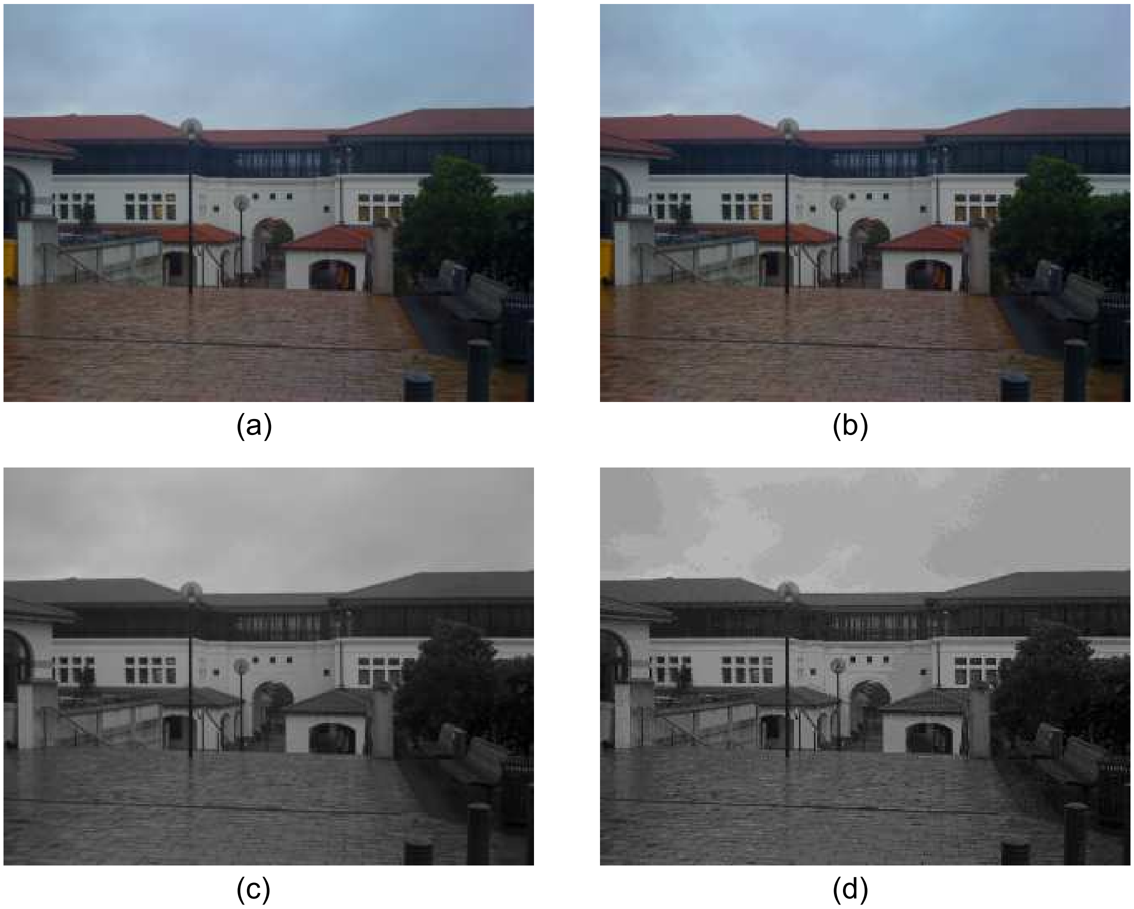

Figure 37: Examples of Colour Depth Reduction; (a) Original 24-bit RGB Image; (b) Reduced 15-bit RGB Image from (a); (c) 8-bit Gray-Scale Image from (a); (d) 4-bit Gray-Scale Image from (c).

knowledge, there is only one similar attempt adhering to the same idea. In [23], a colour depth reduction technique specifically for colour classification is proposed. Their technique however was only partially successful when tested on simple flag identification, stamp identification and landscape classification. Only 3 bits were used to classify a colour pixel and histograms for 8 predefined colour classes (i.e. red, green, blue, cyan, magenta, yellow, white and black) are generated afterwards. Due to lack of colour diversity of some objects being classified, the algorithm may fail as it also does not take into account spatial information.

The algorithms that we employ aim at increasing colour discriminability of the target objects, especially for cases where there are similar colours present in the scene

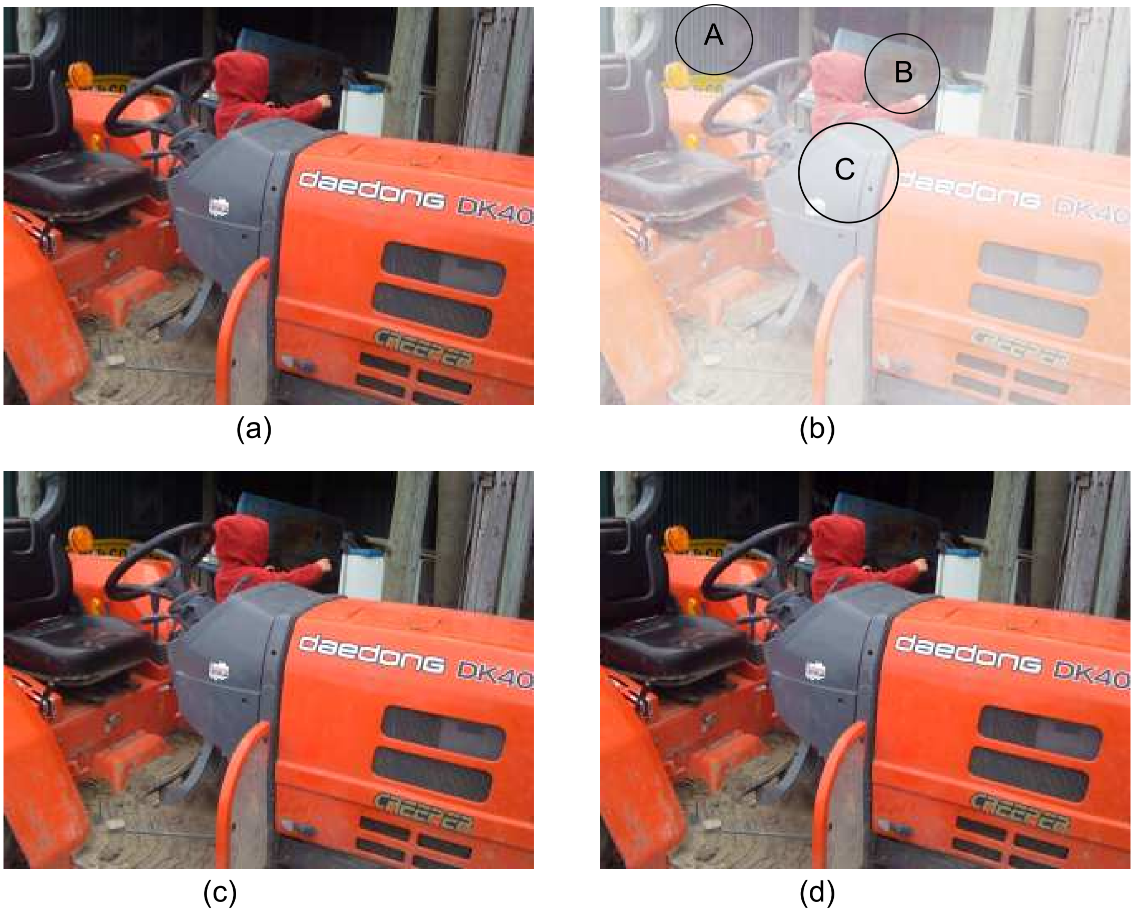

Figure 38: Examples of Colour Depth Reduction; (a) Original 24-bit RGB Image, 55,880 Colours were Used; (b) Areas Where Colour Differences are Visible; (c) 16-bit RGB Image, 4,391 Colours were Used; (d) 15-bit RGB Image 2,814 Colours were Used.

and they need to be classified accurately. For real-time execution, a special Variable Colour Depth LUT is utilised. As a consequence, with the reduced colour depth, the proposed VCD LUT also improves storage efficiency as it requires significantly lesser storage space.

Figure 39, 40, 41, 42, 43, and 44 show graphs of colour contrast enhancement and degradation functions based on three different colour component depth inputs. When the colour component depth decreases, we can observe that there are visible jumps on the output signal because the precision of lower colour component depths is unable to respond more precisely than higher colour component depths.

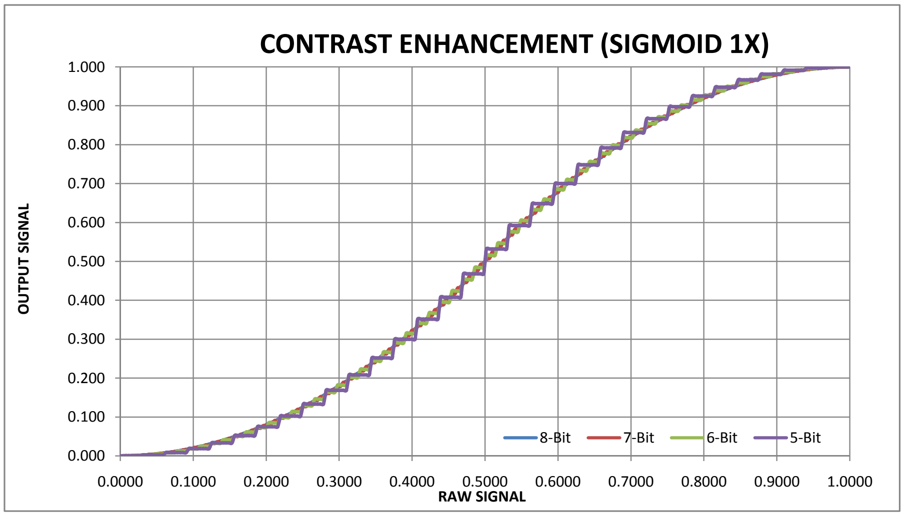

Figure 39: Comparisons of Colour Contrast Enhancement (1X Mode) Results Using Various Colour Depth Representations.

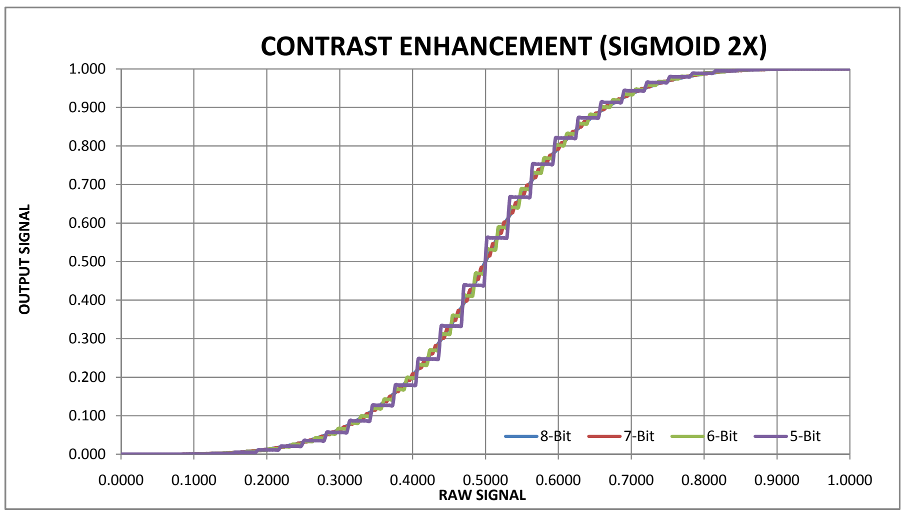

Figure 40: Comparisons of Colour Contrast Enhancement (2X Mode) Results Using Various Colour Depth Representations.

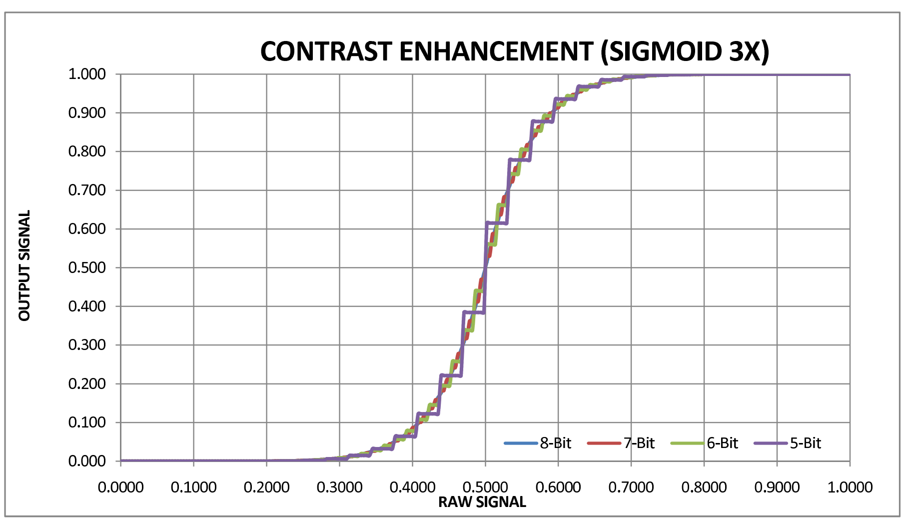

Figure 41: Comparisons of Colour Contrast Enhancement (3X Mode) Results Using Various Colour Depth Representations.

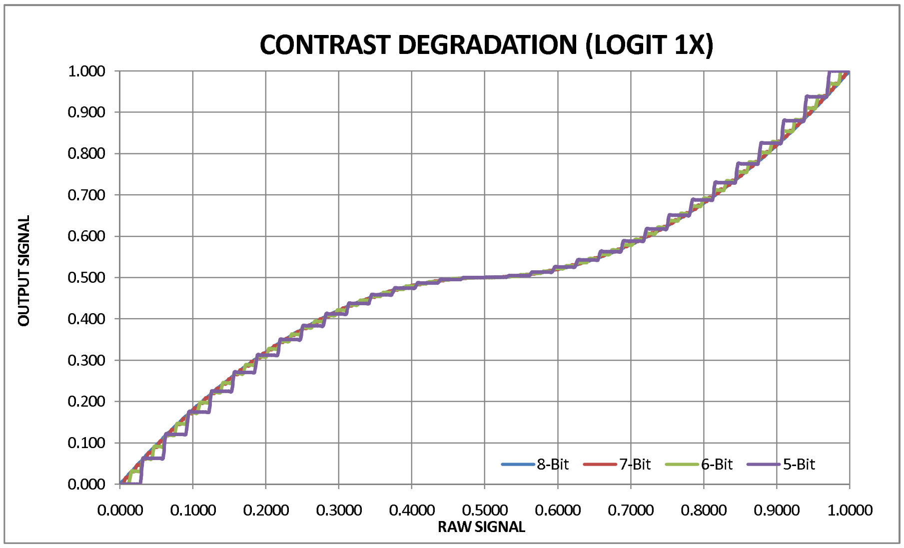

Figure 42: Comparisons of Colour Contrast Degradation (1X Mode) Results Using Various Colour Depth Representations.

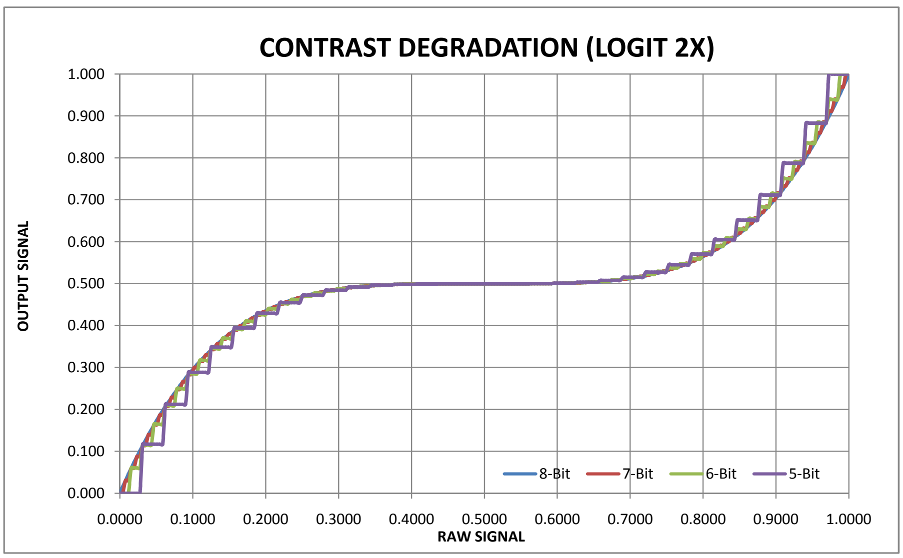

Figure 43: Comparisons of Colour Contrast Degradation (2X Mode) Results Using Various Colour Depth Representations.

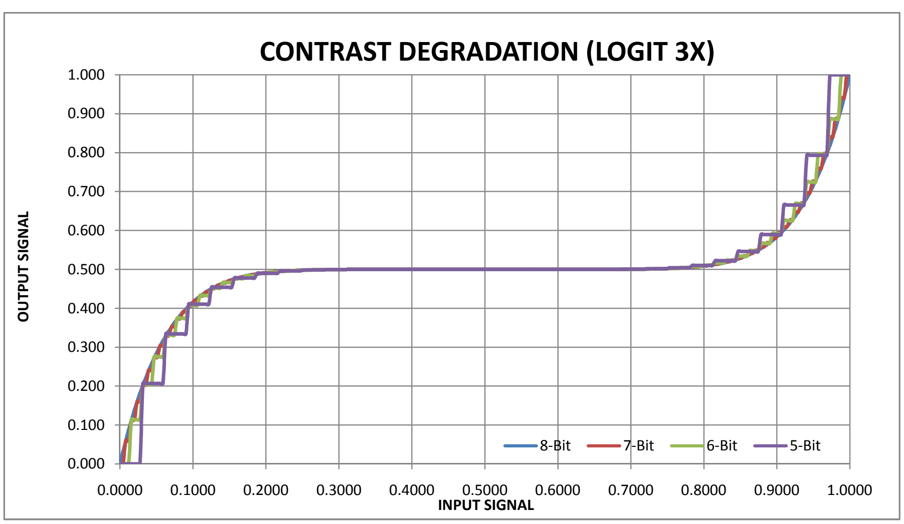

Figure 44: Comparisons of Colour Contrast Degradation (3X Mode) Results Using Various Colour Depth Representations.

The characteristic of these graphs is that the higher colour component depth output is smoother than the jagged lines produced by the lower colour component depth output figure 45. In a reduced colour component depth representation, adap-

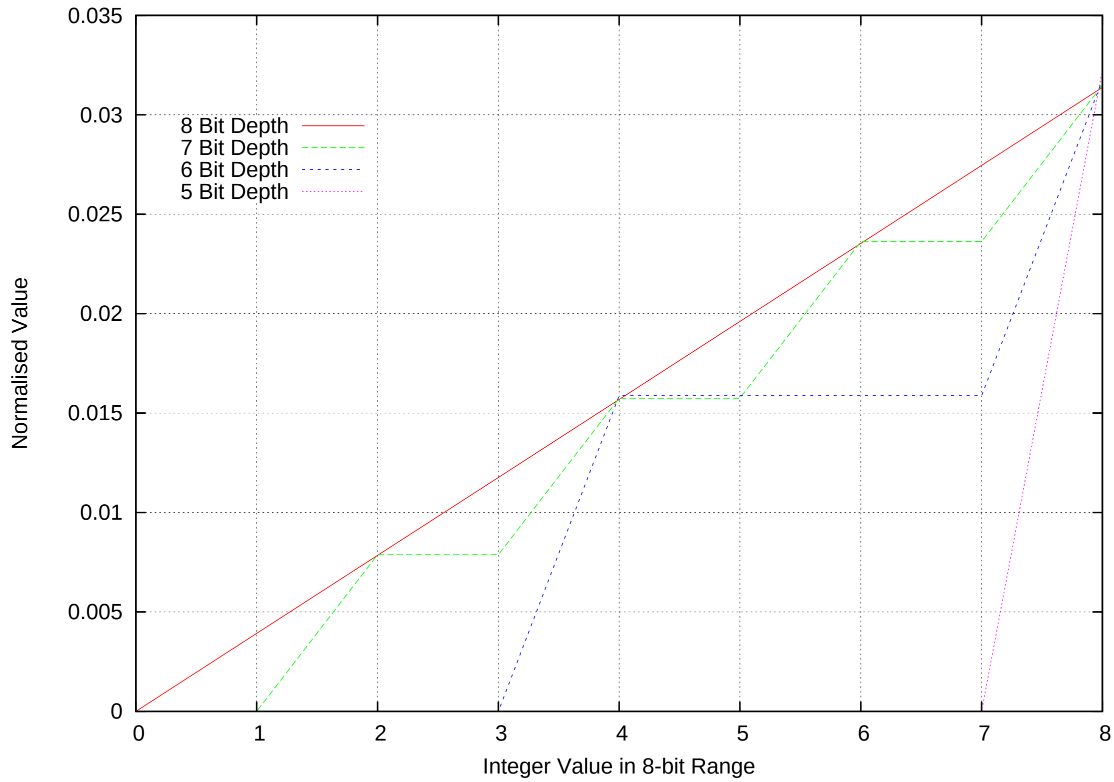

Figure 45: Examples of Normalised Outputs Produced by the Lower Colour Component Depths.

tively varying the colour depths could actually enhance the discriminating features of colours. As an example, as shown in Figure 46, we have represented the red component using only 3-bits, while using 8-bits for both the blue and the green components. It is evident from the graph that reducing the bit representation for the red component allows for better discriminability for those regions in the colour space where all possible shades of red are present. The distinguishable horizontal segmentations appeared in the middle of the colour space where red component ranges from its minimum level up to its maximum level. Thus, for colour classification of target colours that are mostly comprised of the red component (e.g. pink and violet), reducing the red component could actually help improve colour classification accuracy. This is evident in table 8.

Figure 46: RGB Colour Space in Variable Colour Depth of 3-8-8.

4.1.1 Look-up Table (LUT)

A look-up table (LUT) is an array which holds reference values in a pre-defined order. In FCCF, a look-up table is used for fast colour classification. By accessing a look-up table, and indexing it with a given RGB value, the colour classification is determined. In the look-up table building process, a look-up table is constructed by using all possible colour values that could be represented in the colour space. In the RGB colour space, each RGB combination is assigned one of the possible pre-determined colour classes. Therefore, given a colour pixel value (e.g. in RGB), the table is used to determine its colour class.

Standard Indexed LUT

A single look-up table covers the whole colour space, with an assigned colour classifier for each point in the space. An advantage of the standard indexed LUT is that classification can be performed very fast.

Variable Colour Depth LUT (VCD LUT)

The Variable Colour Depth LUT (VCD LUT) differs from the standard indexed LUT as it allows for varying bit numbers in representing each colour component (e.g. 3-bit for red, 8-bits for green and 8-bits for blue). This requires a separate LUT for each pre-defined colour class, but even so, the size of each VCD LUT is very small as it only requires to hold a truth value for referencing a colour value. Altogether, a collection of VCD LUT tables would still be smaller than one standard indexed LUT. To illustrate this, figure 47 shows some comparisons between standard indexed LUT and VCD LUT in terms of LUT structure and memory requirement.

Each entry in the VCD LUT requires only a single bit, whereas the standard

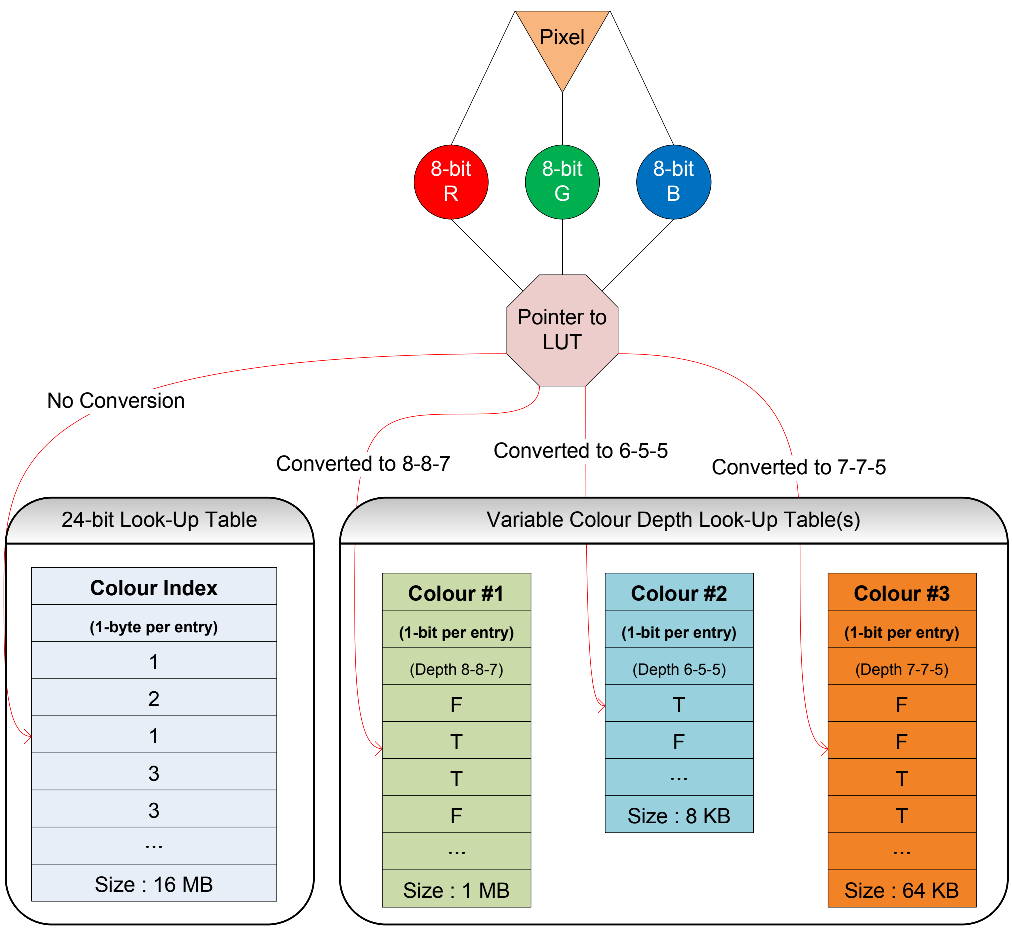

Figure 47: Comparisons between Standard Indexed LUT and VCD LUT.

indexed LUT requires a collection of bits that is sufficient to represent the entire

Table 6: Comparisons of Colour Classification Result between Indexed and VCD LUT

Colour Attributes	**Standard Indexed LUT**	**VCD LUT**		
Red/Green/Blue	Name of Colour	Blue	Violet	Pink
69/5/225	Blue	TRUE	FALSE	FALSE
103/4/217	Blue	TRUE	TRUE	FALSE
133/3/215	Violet	FALSE	TRUE	FALSE
156/1/209	Violet	FALSE	TRUE	TRUE
174/2/207	Pink	FALSE	TRUE	TRUE
196/2/201	Pink	FALSE	FALSE	TRUE
232/1/195	Pink	FALSE	FALSE	TRUE
255/255/255	Undefined	FALSE	FALSE	FALSE

number of colour classes (i.e. a byte for less than 256 colour classes). VCD LUT is suitable for the Variable Colour Depth technique because each LUT is constructed adaptively with varying bit number requirements for each colour component. The colour depth requirements are optimised for each colour classifier. The VCD LUT also renders itself suitable to parallel processing by having exclusive VCD LUT per colour class that could be assigned to an independent process.

Colour Ambiguity and LUT

When a colour value is classified according to multiple classifiers, there is always colour classification ambiguity. Figure 48 shows a situation where colour ambiguity arises due to multiple classifications. The ambiguity may resolved by classifying using neighbouring colour values as cue when ambiguity exists. In the standard indexed LUT however, indications of ambiguities in the colour classification is lost during the construction of LUT because only a single entry of colour classification value is possible. On the other hand, in the VCD LUT, the evidence of ambiguity is indirectly available as multiple LUTs are utilised, indicating different colour classifications for the same colour value. Table 6 illustrates a situation when colour classification ambiguity is present. It also shows how colour classification ambiguity is treated in both standard indexed LUT and VCD LUTs.

Figure 48: Shades of Colour between Blue and Pink.

4.1.2 LUT Building Algorithm

Algorithm 4 builds an LUT for each target colour t, scanning every possible colour values. If the colour value is classified as a target colour, a bit is set in the LUT at a calculated location to indicate membership to that target colour.

4.1.3 LUT Query Algorithm

Given the source colour value of a pixel, Algorithm 5 searches the LUTs of each possible target colours t to classify its colour. The corresponding LUT location for each target colour depth is calculated and a bit mask AND operation is used to extract the target query bit. Note that the LUT location calculation requires shift-left as well as shift-right operations in order to discard excessive bits in the source colour value.

Algorithm 4: Variable Colour Depth LUT Build Algorithm

```
foreach t ← every target n colours do
    for R ← 0 to 2^ddrn − 1 do              // Every possible Red values
        for G ← 0 to 2^ddgn − 1 do          // Every possible Green values
            for B ← 0 to 2^ddbn − 1 do      // Every possible Blue values
                if colour value is classified as colour t then
                    L = (R≪(ddgn+ddbn))+(G≪ddbn)+B;
                    LB = L ≫ (b log2);
                    Lb = 1 ≪ (L mod b);
                    LUT[t][LB] = LUT[t][LB] ⋃ Lb;
```

dsr, *dsg*, *dsb* are the colour depth values of each colour channel in source image
ddrn, *ddgn* and *ddbn* are the colour depth values of each colour channel in each target colour LUT
LB is an index to the LUT that corresponds to the target colour
b is the size of data type of LUT optimised for the system architecture (e.g. 8 for byte-aligned, and 16 for word-aligned)

4.1.4 General Variable Colour Depth - FCCF System Architecture

Figure 49 illustrates how the VCD LUT is constructed per colour classifier. All possible colour values in a given Variable Colour Depth is tested in the FCCF process to construct the VCD LUT.

Figure 50 shows how the VCD LUT is used in a real-time environment. The acquired colour pixel in the scene is converted into a separate Variable Colour Depth representation for each colour classifier. Next, each colour classifier accesses its own VCD LUT to determine the pixel's colour class. If there is only a single target colour object to track, then only one colour classifier is required to test, along with its own VCD LUT.

Algorithm 5: Variable Colour Depth LUT Query Algorithm

foreach $t \leftarrow$ *every target n colours* **do**
- R = Red component value of target pixel;
- G = Green component value of target pixel;
- B = Blue component value of target pixel;
- $L = ((R \gg (dsr\text{-}ddrn)) \ll (ddgn+ddbn))+((G \gg (dsg\text{-}ddgn)) \ll ddbn)+(B \gg (dsb\text{-}ddbn))$;
- $LB = L \gg (b \log_2)$;
- $Lb = 1 \ll (L \bmod b)$;
- **if** *(*LUT[t][LB] $\bigcap$ Lb*)* $\neq$ *0* **then**
  - Given pixel is qualified for target colour t

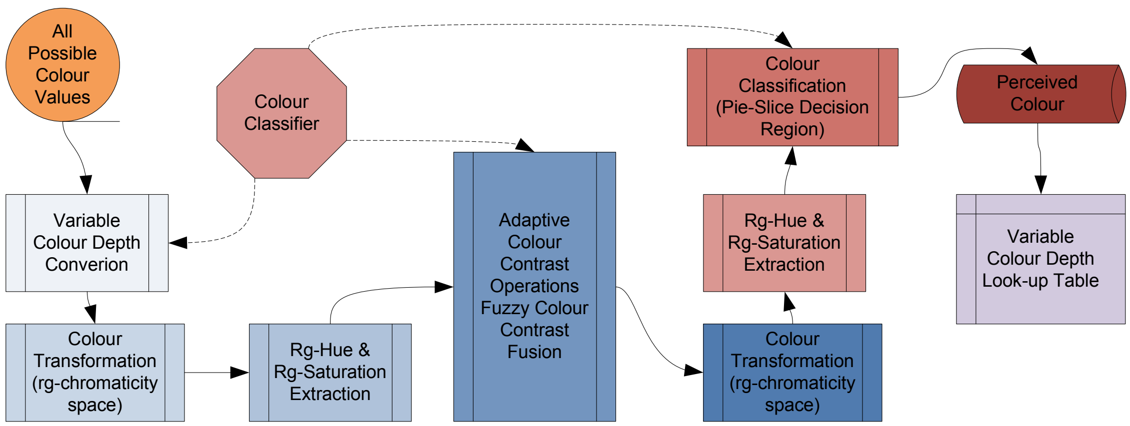

Figure 49: Variable Colour Depth Look-Up Table Construction Architecture

4.2 Fuzzy-Genetic Colour Classifier Search

4.2.1 Motivation

Genetic Algorithm is considered to be a non-exhaustive search technique suitable for finding optimal or near optimal solutions for any given problem domain. The Fuzzy-Genetic Colour Calibration experiments designed in this research aims to find optimal parameter sets for accurate colour classification. However, the search space to be explored in finding an optimal colour classifier in FCCF is vast due to the real number valued-parameters of classification, such as contrast angles and radii to mention a few. Although the radii may be discretised to define the search space, angles are difficult to quantise due to the inherent characteristic of the arc length

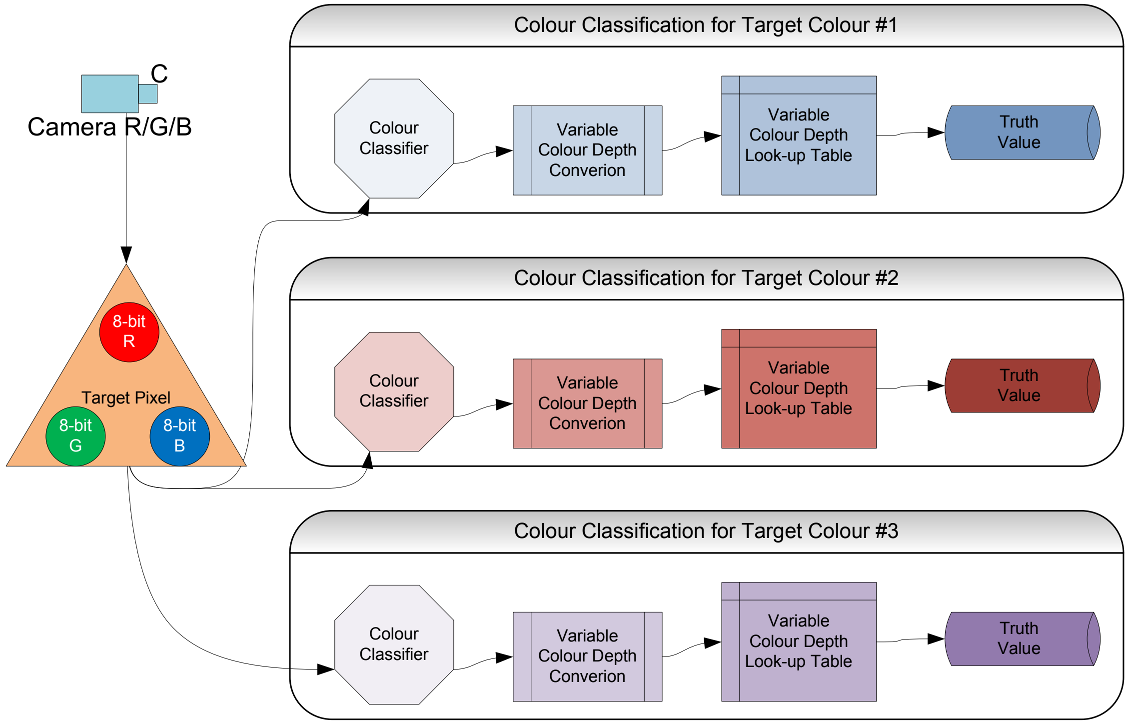

Figure 50: Variable Colour Depth Look-Up Table for Real-Time Processing

that is a product of angle and radius as shown in figure 51 and equation 15. Therefore, when the angles are discretised, for example, covering an area represented as $A, b, andc$ in figure 51, the gradations as a result of discretisation will eventually cause some inaccuracy. The inaccuracy of angle representations are represented as gray areas in *bandc*, and they also represent the region of errors. When the angle inaccuracy increases, this consequently enlarges the region of errors significantly. On the other hand, when the radius inaccuracy increases, the region of errors grows proportionally, but with lesser effect than the increase in angle inaccuracy.

$$L = \theta r \tag{15}$$

The colour classifier requires a large number of parameters to calibrate. This leads to a lot of difficulty in generating an optimal colour classifier automatically. There

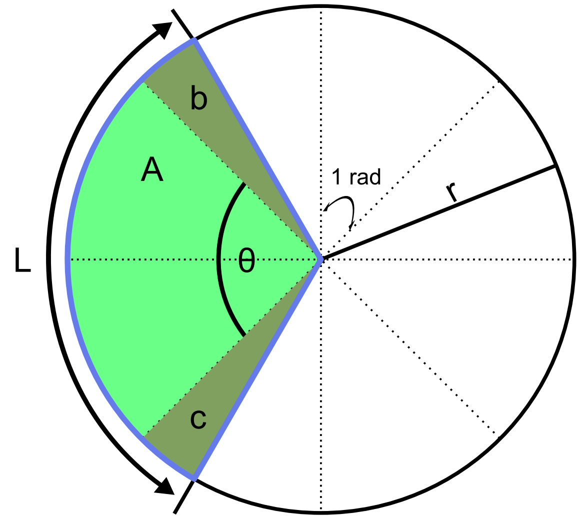

Figure 51: The Shaded Pie-Slice, Covered by Arc L Represents The Area of Interest. Due to the Discretisation of the Angles, the Area of Interest Could Only be Approximated by a Smaller Pie-Slice, Covered by Angle θ. The Dotted Lines Indicate the Discretisation of Angles.

are 6 real number value parameters (classification angles, contrast angles and classification radii), 3 sets of classification operations and 3 sets of Variable Colour Depth subranges which all affect the result of colour classification independently. Genetic Algorithm offers to find a solution from the search space by performing mutation and crossover operations on the chromosomes. The algorithm may end up with a non-optimal solution. However, it is highly likely to return a more accurate set of colour classifier parameters than the manually calibrated ones.

Furthermore, it is also possible to feed a previously discovered solution set to the Genetic Algorithm repeatedly to allow it to evolve closer to the optimal colour classifier, however due to the size and complexity of the colour parameter optimisation problem, this approach is not exactly suitable to the problem at hand. It was empirically observed in this research that starting anew with a different random seed returns better results when the previous results was not satisfactory. These conditions can be viewed in 5.3.

4.2.2 General Architecture

Figure 52 illustrates the overall architecture of Fuzzy-Genetic colour classifier search architecture. The Genetic Algorithm parameters and reference image are fed into the Genetic Algorithm to generate a colour classifier.

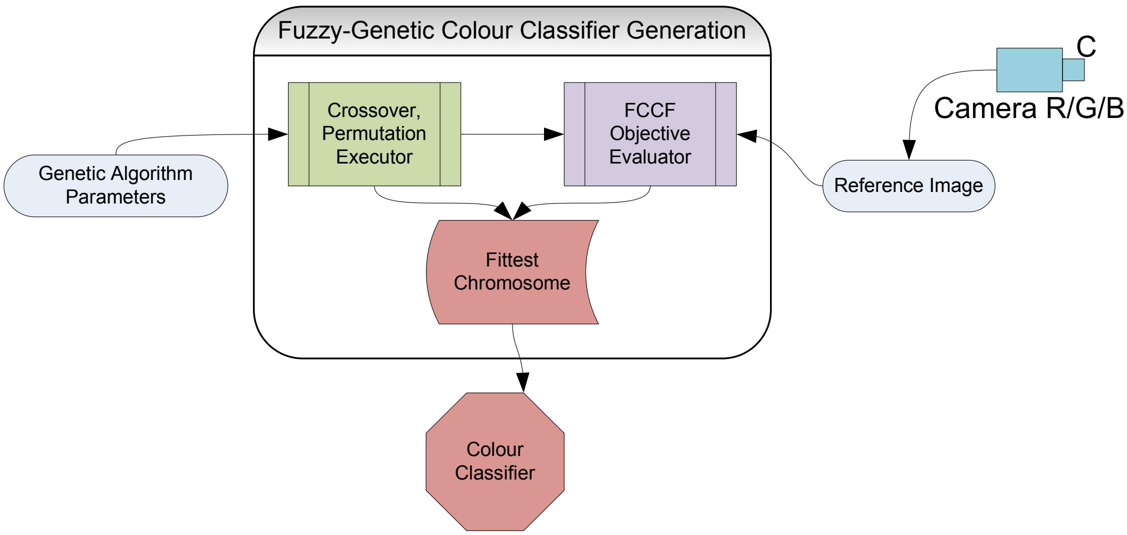

Figure 52: Fuzzy-Genetic Colour Classifier Search Architecture

4.2.3 Chromosome Design

The chromosome defines the search space of the Genetic Algorithm. As discussed in the earlier sections, there is a total of 12 different parameters to construct a colour classifier; thus, there are also 12 different parameters to optimise. We designed the chromosome with a total size of 108 bits (figure 53). The chromosome design is largely divided into two sections. The front 60 bits correspond to angles and radii, while the last 48 bits correspond to contrast rules and colour depth values. We divided it into two because the latter 48 bits could be disabled when using the guided search strategy, which is explained shortly thereafter.

For the 60 bits front part of the chromosome, 4 angles and 2 radii are assigned each with a 10-bit range sub-chromosome. Each sub-chromosome could represent 2^{10}

values, which ranges from 0 to 1 representing the radius. This is about 0.001 incremental steps for the radius. If the angle parameter is using the full 0 to 360 range, the increments are about 0.35 degree. However, this can be sliced more narrowly if we limit the search range for the angles. In the experiments, we limited the search range for the angle up to 180 degrees. This allows incremental steps of about 0.176 degree.

The last 48 bits of the chromosome, divides into a length of 8-bits for representing the integral values of the contrast rules and colour depth. Since 8 bits are somewhat larger than the required 7 states of colour contrast rules and 4 levels of colour depths, it allows for larger variances of crossover and mutation operations.

Parameter	Range	Length	Incremental Steps
Min Angle	Pivot -30° to -90°	10 bits	0.058 ~ 0.176
Max Angle	Pivot +30° to +90°	10 bits	0.058 ~ 0.176
Min Radius	0 ~ 1	10 bits	0.001
Max Radius	0 ~ 1	10 bits	0.001
Min Contrast Angle	Pivot -30° to -90°	10 bits	0.058 ~ 0.176
Max Contrast Angle	Pivot +30° to +90°	10 bits	0.058 ~ 0.176
Red Contrast Rule	-3.00 to 3.99	8 bits	0.027
Green Contrast Rule	-3.00 to 3.99	8 bits	0.027
Blue Contrast Rule	-3.00 to 3.99	8 bits	0.027
Red Colour Depth	5 to 8.99	8 bits	0.015
Green Colour Depth	5 to 8.99	8 bits	0.015
Blue Colour Depth	5 to 8.99	8 bits	0.015

Figure 53: Chromosome Design

4.2.4 Fitness Function

The fitness function, also known as the objective function gives fitness values that represent the ranks of chromosomes evaluated during the optimisation process. The fitness of the chromosomes tells exactly how close is the generated solution to the goal is. The fitness function used for the Fuzzy-Genetic colour classifier search employs the colour classification scoring formula proposed in [11] (Algorithm 3). The scoring function awards 1.0 for a perfect colour classification and is totally independent of the structure and/or number of colour classifier parameters. This is a very desirable feature for a fitness function.

4.3 Summary

We have presented a new concept called Variable Colour Depth and discussed its characteristics which may improve colour classification accuracy. The supporting algorithms for Variable Colour Depth are also introduced, such as build and search algorithms and VCD LUT for solving colour ambiguity problems and boosting its performance. We have also proposed use of Genetic Algorithm to find better colour classifiers, and introduced a chromosome design for the task.

In the next chapter, we discuss an experiment conducted on the Variable Colour Depth algorithm to test its efficacy. We also test the Fuzzy-Genetic colour classifier and search strategy.

Chapter 5

Experiments and Analysis

The experiments were performed on the same robot soccer test bed used in [11] for comparison purposes. However, the calibration set up is non-typical, as it is plagued with spatially varying illumination intensities, with 6 target colours, represented by 40 colour patches, strategically positioned to be exposed under different illumination conditions (i.e. dim, dark, bright). The focus of the experiments is divided into two parts. One is to compare colour classification results when the full colour depth (24 bits) is used vs. Variable Colour Depth and the latter part compares the best colour classification results from Variable Colour Depth with brute force Colour Contrast Rule Extraction (CCRE) against results from Genetic Algorithm generated colour classification parameters.

5.1 Test Setup

5.1.1 Assessment Method

The classification performance is gauged based on a scoring formula proposed in [11]. Colour Contrast Rule Extraction (CCRE) which reviewed in chapter 3 (Algorithm 3) describes how the scoring formula is constructed. The formula takes into account the number of true positives, false positives, as well as the area of the target colour objects, and has proven to identify the superior colour contrast rule combination.

5.1.2 Reference Result

Table 7 is extracted from the results of a previous research [11]. The angles and radii were hand-calibrated and colour contrast rules were automatically extracted by a brute force search method [11]. This experiment sets a goal that any result better than this reference result is satisfactory.

Table 7: Colour Classification Definition

Base parameters							Extracted rules and results			
Colour	Angle		Radius		Contrast Angle		Contrast Rule	Score	Hits	Misses
	Min	Max	Begin	End	Min	Max	R , G ,B			
Yellow	43.992	46.476	0.003	1	41.832	47.808	3, 1, -2	0.648530	2104	68
Green	45.576	96.66	0.0547	1	45.288	96.064	0, -1, -3	0.552422	3313	383
Pink	314.424	327.276	0.1461	1	275.256	331.884	1, -1, 0	0.586446	1714	99
Purple	286.524	307.859	0.13	1	285.012	308.556	0, 1, -3	0.572888	2777	314
Violet	232.344	282.276	0.044	1	228.312	293.364	1, 1, 2	0.526654	2535	497
Light Blue	137.124	162.792	0.019	1	136.944	163.512	0, 3, 1	0.668808	2758	68

5.2 Variable Colour Depth with CCRE

This experiment tests the effectiveness of using VCD by applying an extended version of CCRE which includes a new parameter (i.e. variable colour depth) for evaluating the performance. If the experiments give us more accurate classification results from a colour depth representation less than the full 24-bit colour depth, then that would suggest that some colour components are less important than others for classifying target colours. Consequently, this would also prove that FCCF have the capability of compensating for loss of colour component resolution.

5.2.1 Search Strategy

In order to find alternative colour depth values, a brute force search method was employed. Each candidate target colour classification holds the base parameters (i.e. angles, radii) retrieved from the previous results [11]. These parameters were kept constant, while the colour contrast rules and colour depth values were permutated to find the most accurate colour classification parameters. We limit the colour depth search space to only 64 possibilities. In effect, we considered only from 5-bits to 8-bits, per colour component. It is deemed that colour depth representations less than 5-bits per component wouldn't provide enough resolution for effective segmentation. It is also too costly to search to consider all 8^3 possibilities. For each target colour depth, there are 343 different colour contrast rules to test; therefore, there will be 21,952 colour classification tests required per target colour.

5.2.2 Colour Classification Results of Full 24-bit Colour Depth vs. Variable Colour Depth

We employed the same colour classification definition for the 6 target object colours tested in [11] for direct comparison of algorithm performances. The previous research used a 24-bit colour depth LUT for each target colour, and utilised an algorithm for automatic extraction of the angles and radii values, and colour contrast rules. Table 8 shows comparisons between the best scores from the previous research and this research.

As observed from the table, it is clear that the application of the Variable Colour Depth approach resulted to better scores than the full 24-bit colour depth LUT in all 6 target object colours. It is evident that the misclassifications have been significantly reduced down for all target colours.

5.2.3 Colour Contrast Rules and Scores

Table 8 details the results of the experiments on optimised colour contrast rule extraction at varying colour depth. The table reflects the scores garnered by the rule combinations in classifying the 6 target colours (represented by 40 colour patches). The table indicates the colour depth and colour contrast operation used for each of the colour channel, the performance of the rule combination in terms of the number of hits, misclassifications, storage space requirement, improvement over the full-colour-depth LUT and rule combination's relative ranking. The best results show that FCCF increased colour classification even when there were lost bits in the colour depth.

Table 8: Colour Classification Results of Full 24-bit Colour Depth vs. Variable Colour Depth

Colour	Angle Min, Max	Depth R , G ,B	Contrast Rule R ,G ,B	Score	Hits	Misses	LUT Size	Improvement Rate	Rank
Yellow	43.992 46.476	8, 8, 8	3, 1, -2	0.648530	2104	68	2048KB	0%	3
		7, 8, 6	1, 3, -2	0.655979	2261	96	256KB	1.149%	1
		6, 8, 6	-1, 1, -3	0.609815	2258	172	128KB	-5.97%	4
		7, 8, 8	1, 3, -2	0.655374	2261	97	1024KB	1.055%	2
Green	45.576 96.66	8, 8, 8	0, -1, -3	0.552422	3313	383	2048KB	0%	4
		6, 5, 8	1, 0, -2	0.639059	3137	127	64KB	15.683%	1
		6, 8, 6	2, 1, -2	0.587168	3266	288	128KB	6.29%	3
		7, 8, 8	0, -1, -3	0.615805	3239	206	1024KB	11.474%	2
Pink	314.424 327.276	8, 8, 8	1, -1, 0	0.586446	1714	99	2048KB	0%	4
		7, 8, 7	1, -1, 0	0.622773	1679	46	512KB	6.194%	1
		6, 8, 6	1, -1, 0	0.603303	1623	58	128KB	2.874%	3
		7, 8, 8	1, -1, 0	0.616230	1684	55	1024KB	5.079%	2
Purple	286.524 307.859	8, 8, 8	0, 1, -3	0.572888	2777	314	2048KB	0%	2
		6, 7, 7	0, 1, -3	0.576178	2782	309	128KB	0.574%	1
		6, 8, 6	0, 1, -3	0.565163	2729	313	128KB	-1.348%	4
		7, 8, 8	0, 1, -3	0.572094	2773	314	1024KB	-0.139%	3
Violet	232.344 282.276	8, 8, 8	1, 1, 2	0.526654	2535	497	2048KB	0%	4
		5, 7, 5	0, 1, 1	0.602979	1802	101	16KB	14.492%	1
		6, 8, 6	1, 1, 2	0.545970	2502	442	128KB	3.668%	2
		7, 8, 8	0, 0, 2	0.529645	2400	425	1024KB	0.568%	3
Light Blue	137.124 162.792	8, 8, 8	0, 3, 1	0.668808	2758	68	2048KB	0%	4
		5, 5, 6	0, 3, 1	0.690887	2786	30	8KB	3.301%	1
		6, 8, 6	0, 3, 1	0.671966	2703	48	128KB	0.472%	2
		7, 8, 8	0, 3, 1	0.671255	2758	63	1024KB	0.366%	3

5.2.4 Colour Contrast Rule Clustering

The clustering of the best set of colour contrast rules for each possible colour depth value, and for each target colour is shown in figure 54. The colour depth values were varied from 5 to 8, considering a total of 64 possible permutations inspected for each target colour. The actual numeric figures are presented in Table 9.

On the other hand, figure 55 shows a mapping of the best colour contrast rule combinations for the optimal colour depth values for each target colour. It also reflects the number of occurrences of the same colour contrast rule combination for each target colour. For example, *Light Blue (0/3/1)*59* from figure 55 indicates the following colour contrast rule: no contrast on red, enhance three times on green and enhance once on blue component. This also corresponds to the best colour contrast rule for 59 colour depth combinations out of 64. As indicated in the figures, the optimal colour contrast rule combinations for Light Blue, Pink and Purple adhere with the majority of colour contrast rule combinations found at different colour depth values. There are some observable patterns in the rule combinations as well. For instance, Green always requires the colour contrast degrade operation on the Blue channel. (Table 9) In contrast, there is no observable pattern on Yellow's contrast rule combinations; they are scattered all over the rule space.

Table 9: Colour Contrast Rule Distribution

Levels		Yellow			Green			Pink		
		Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
	-3	0	1	7	0	0	29	0	0	9
Degrade	-2	9	7	11	0	1	35	0	0	2
	-1	29	21	4	11	23	0	4	52	13
Neutral	0	4	3	4	13	21	0	0	2	40
	1	10	14	19	19	9	0	50	3	0
Enhance	2	6	2	1	21	6	0	3	0	0
	3	6	16	18	0	4	0	7	7	0

Levels		Purple			Violet			Light Blue		
		Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
	-3	0	0	48	0	0	0	0	0	0
Degrade	-2	0	0	16	0	0	0	0	0	0
	-1	0	0	0	0	0	0	0	0	0
Neutral	0	64	0	0	50	23	0	59	0	0
	1	0	64	0	14	41	26	5	0	64
Enhance	2	0	0	0	0	0	38	0	0	0
	3	0	0	0	0	0	0	0	64	0

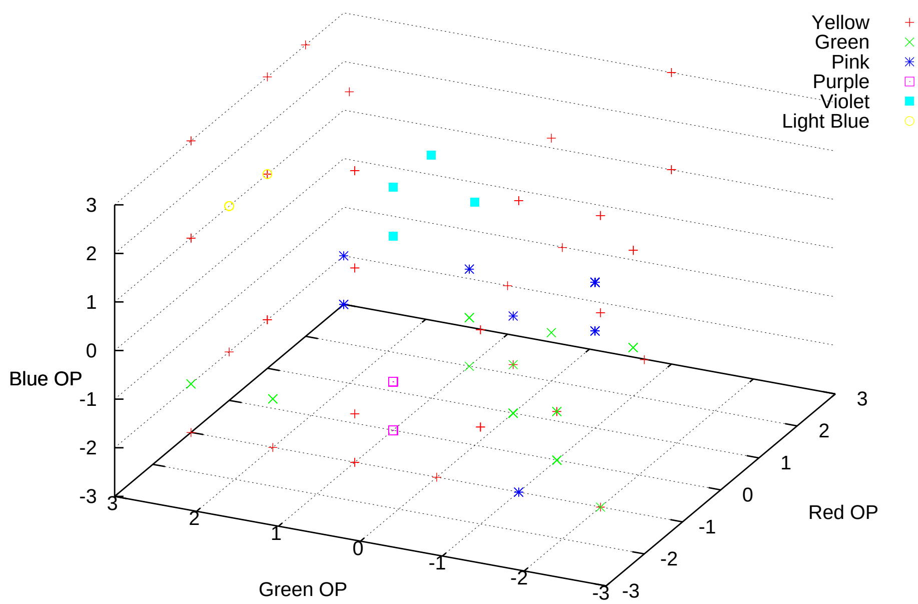

Figure 54: Mapping of all the Best Colour Contrast Rule Combinations for all Colour Depth Values and for each Target Colour

5.2.5 Colour Pixel Clustering

Figure 56 shows 2 sets of data collected from 8 Light Blue objects under different illumination intensities. These data plots represent the colour pixels of the objects in the rg-Hue vs. rg-Saturation chart. The first set was generated using a colour depth of 8-8-8 bits (8-bits for the red component, 8-bits for the green component and 8-bits for the blue component) denoted by '+', while the second set was generated using a colour depth of 5-5-6 bits denoted by 'x'. Most of the pixels are clustered within the minimum and maximum pie-slice decision angles of 137.124 to 162.792, and radii between 0 to 0.1. It can also be observed that the lower colour depth pixels denoted by 'x' relatively spread evenly across the bounding angle's due to loss of colour resolution.

Figure 57 is closely related to Fig. 56 as it shows the clustering of pixels of the same target objects in Fig. 56 enlarged at pie-slice decision angles, with the same illumination intensities and colour depths, except that the FCCF algorithm was

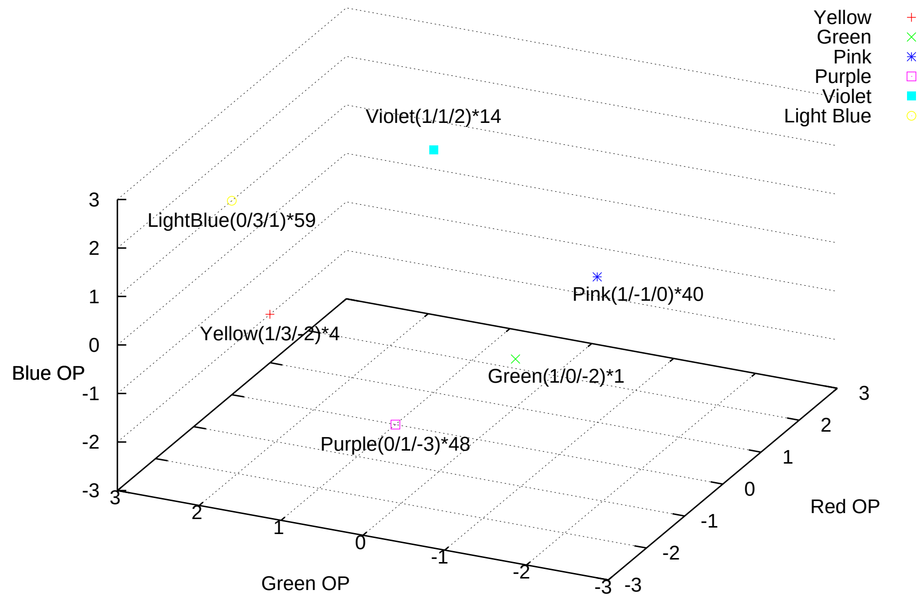

Figure 55: Mapping of the Best Colour Contrast Rule Combinations for the Optimal Colour Depths for each Target Colour. Positive Number Indicates Contrast Enhancement and Level of Contrast Application; 0 for No Operation, while a Negative Number Denotes Contrast Degradation. * n Indicates Number of Occurrences.

applied. 2 sets of data were collected. The first set was generated using a colour depth of 8-8-8 bits (denoted by '+', while the second set was generated using a colour depth of 5-5-6 bits denoted by 'x'. For the 2 data sets, the following colour contrast rules were applied: Red channel: no operation; Green channel: enhance 3 times; Blue channel: enhance 1 time. When FCCF was applied, it can be observed that the colour pixels close to the maximum angle, 162.792 were pulled inside the pie-slice decision region and were spread toward covering a broader radius. In effect, the lower colour depth pixels are now clustered with some regularity.

Further experiments show evidences that FCCF improves colour classification of other target colours by influencing the formation of the colour pixels within the confines of the pie-slice decision region. Other colour clustering results can be observed from Figure 58, 60, 62, 64, 66, and 69.

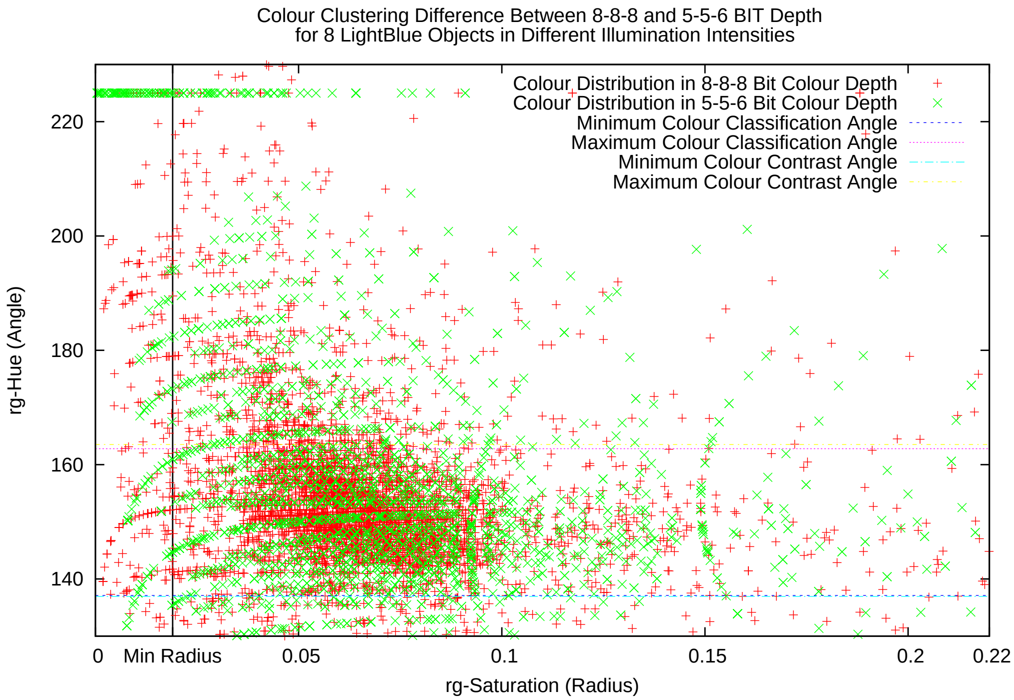

Figure 56: Colour Pixel Clustering on rg-Hue / rg-Saturation Chart for Light Blue Objects

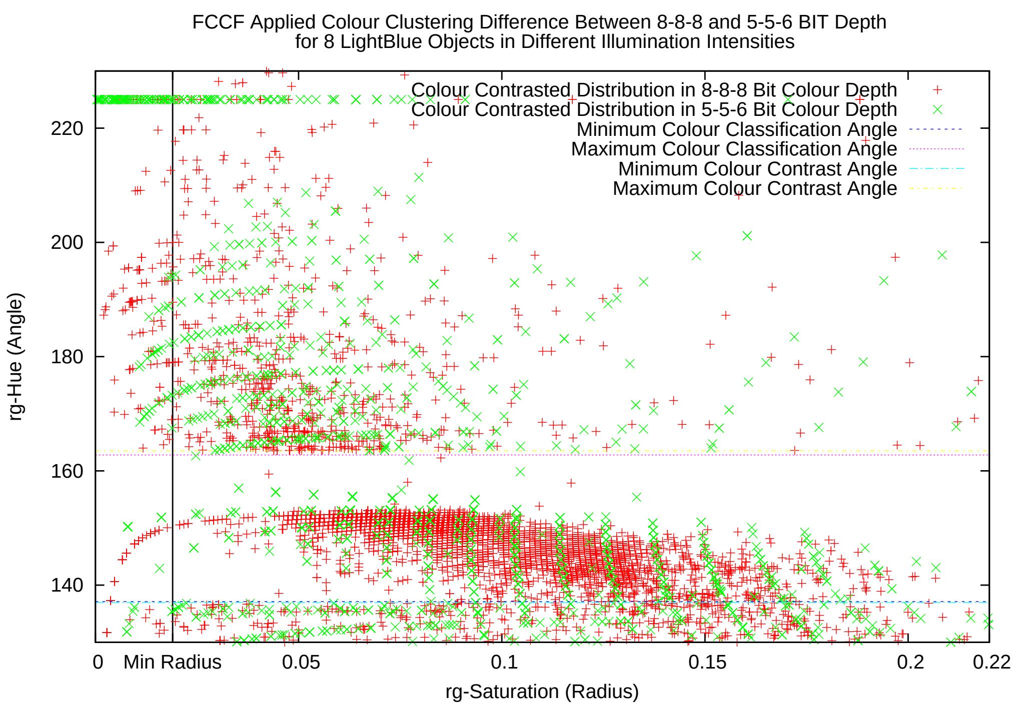

Figure 57: Colour Pixel Clustering on rg-Hue / rg-Saturation Chart for Light Blue Objects with FCCF

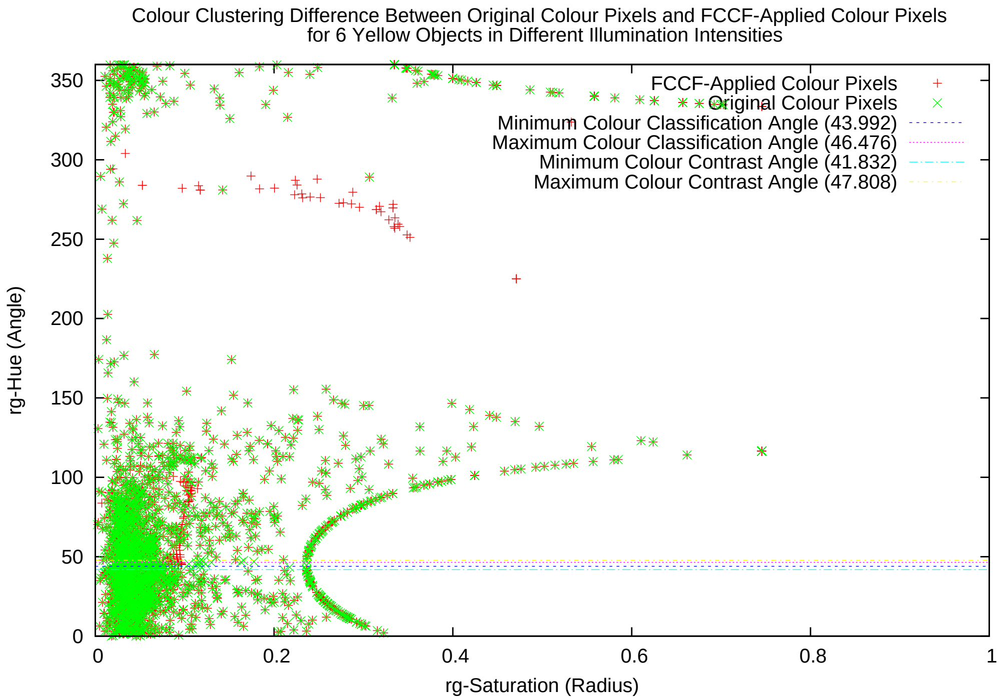

Figure 58: Colour Pixel Clustering on rg-Hue / rg-Saturation Chart for Yellow Objects

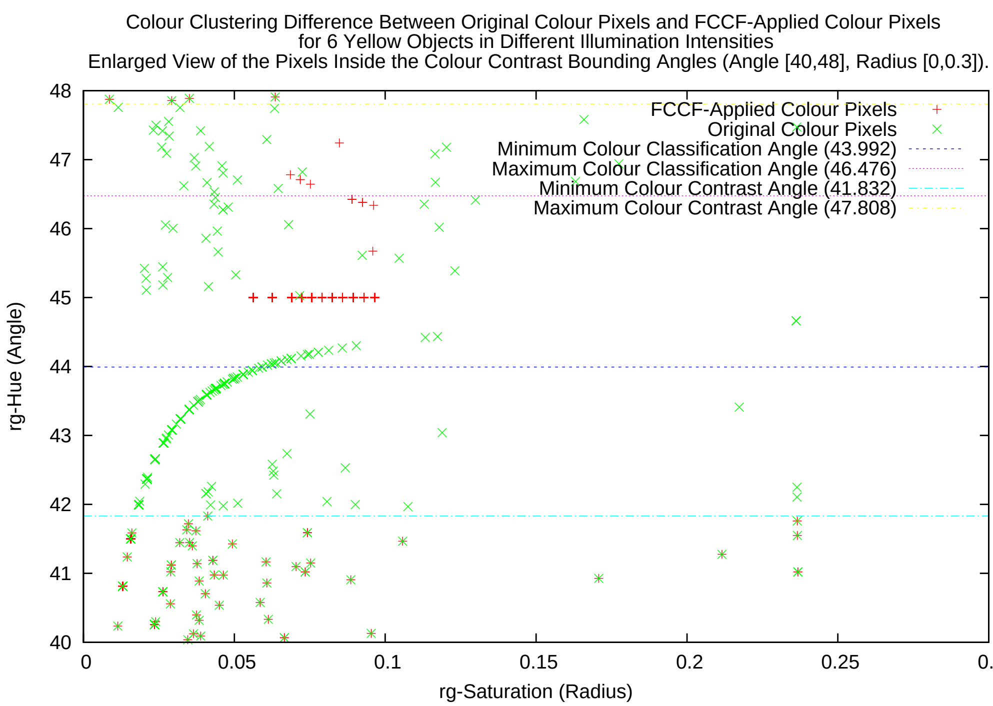

Figure 59: Enlarged Colour Pixel Clustering on rg-Hue / rg-Saturation Chart for Yellow Objects

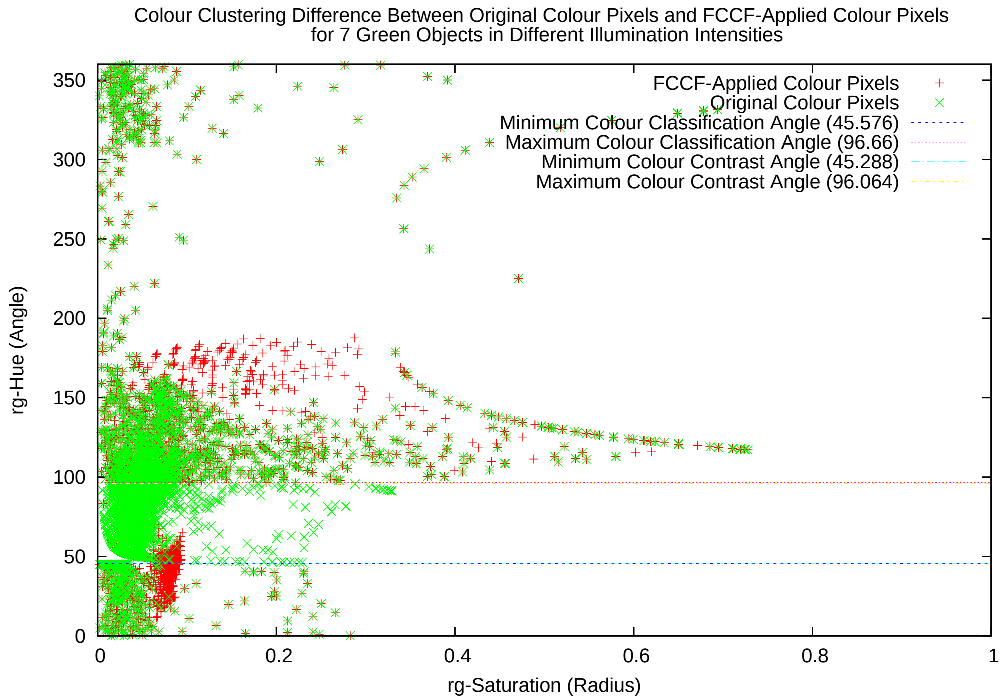

Figure 60: Colour Pixel Clustering on rg-Hue / rg-Saturation Chart for Green Objects

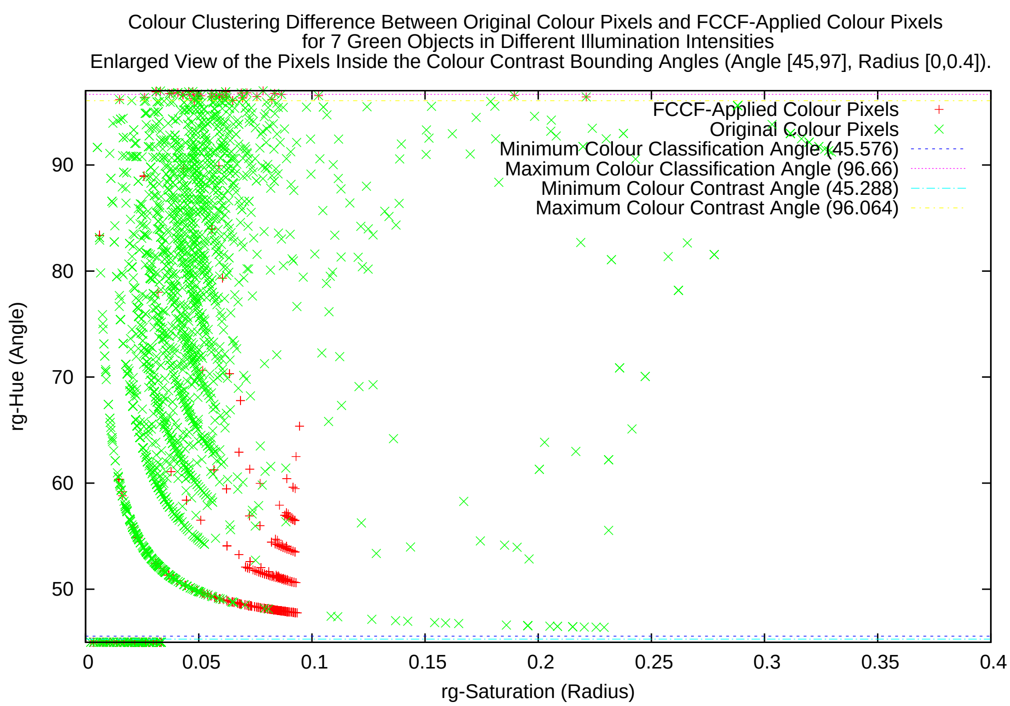

Figure 61: Enlarged Colour Pixel Clustering on rg-Hue / rg-Saturation Chart for Green Objects

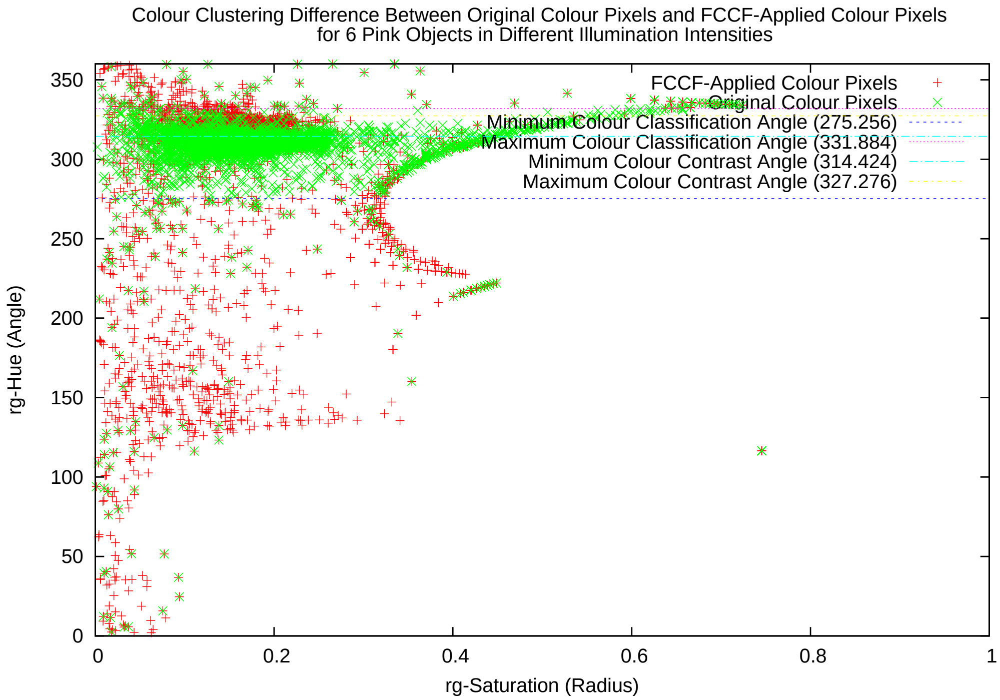

Figure 62: Colour Pixel Clustering on rg-Hue / rg-Saturation Chart for Pink Objects

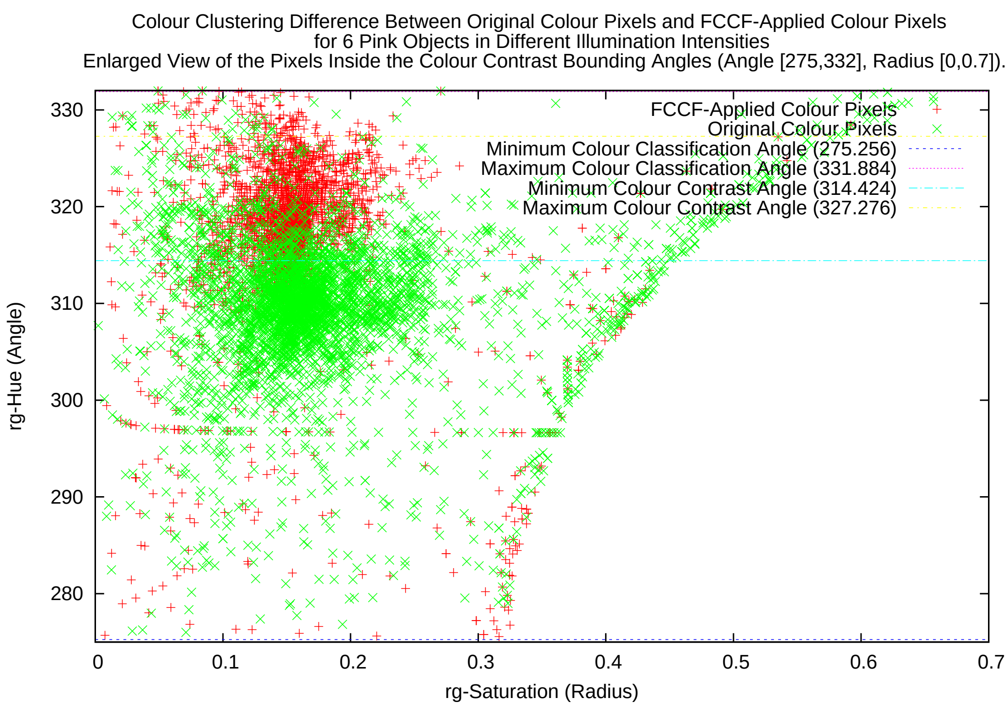

Figure 63: Enlarged Colour Pixel Clustering on rg-Hue / rg-Saturation Chart for Pink Objects

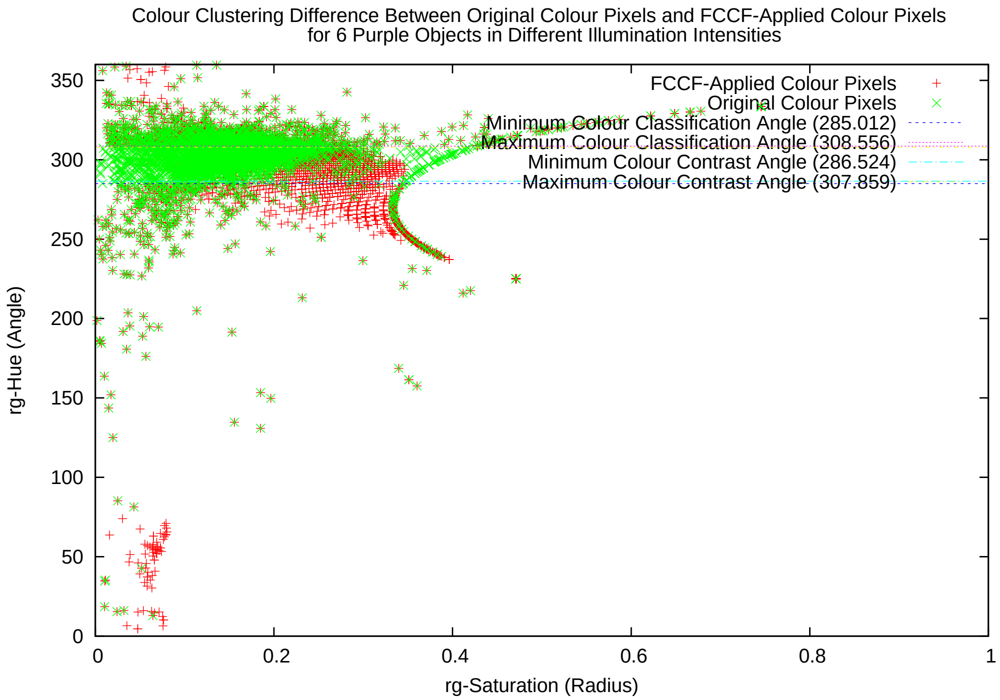

Figure 64: Colour Pixel Clustering on rg-Hue / rg-Saturation Chart for Purple Objects

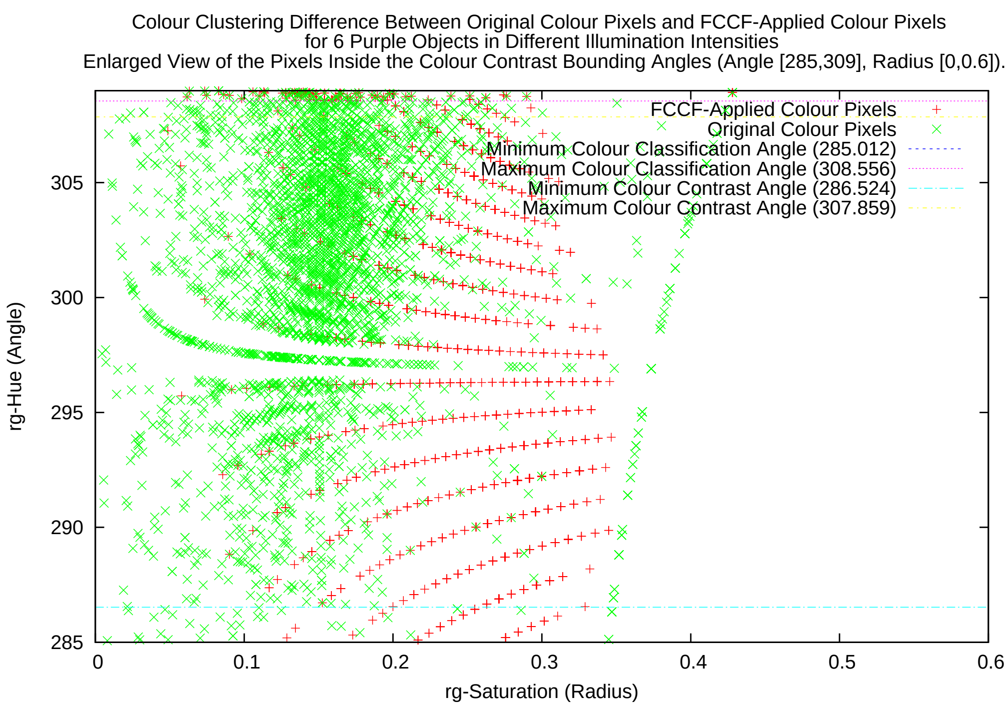

Figure 65: Enlarged Colour Pixel Clustering on rg-Hue / rg-Saturation Chart for Purple Objects

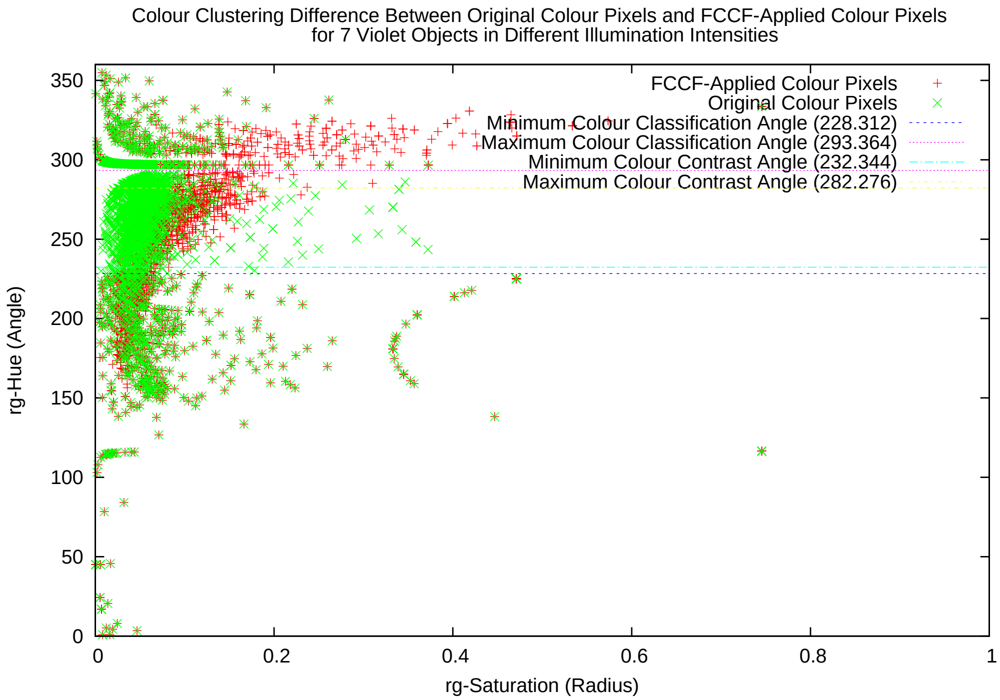

Figure 66: Colour Pixel Clustering on rg-Hue / rg-Saturation Chart for Violet Objects

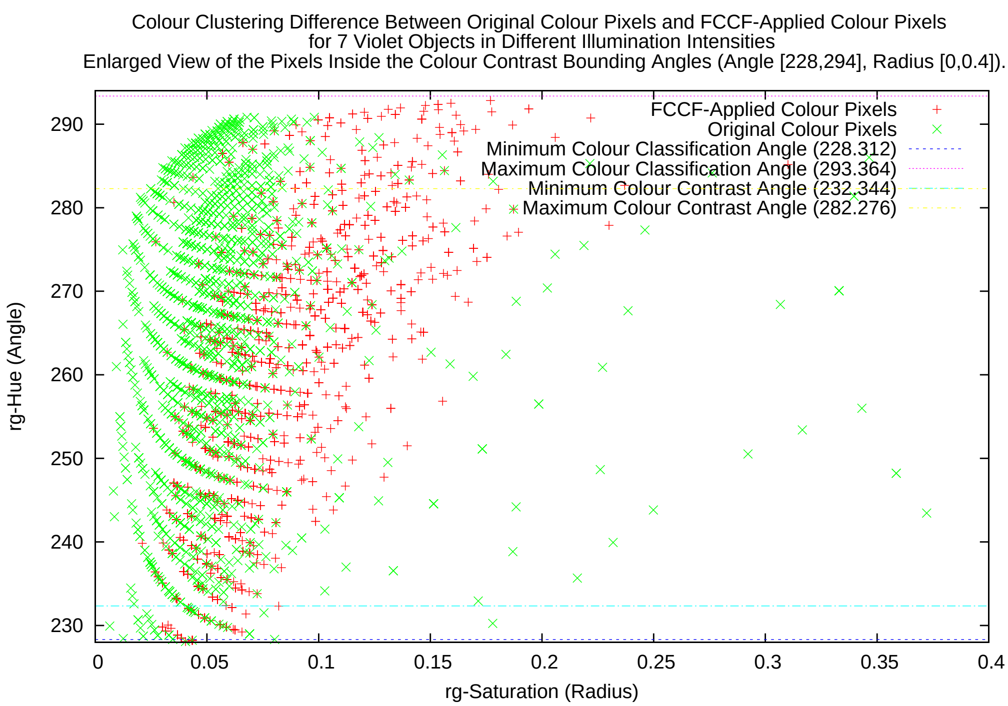

Figure 67: Enlarged Colour Pixel Clustering on rg-Hue / rg-Saturation Chart for Violet Objects

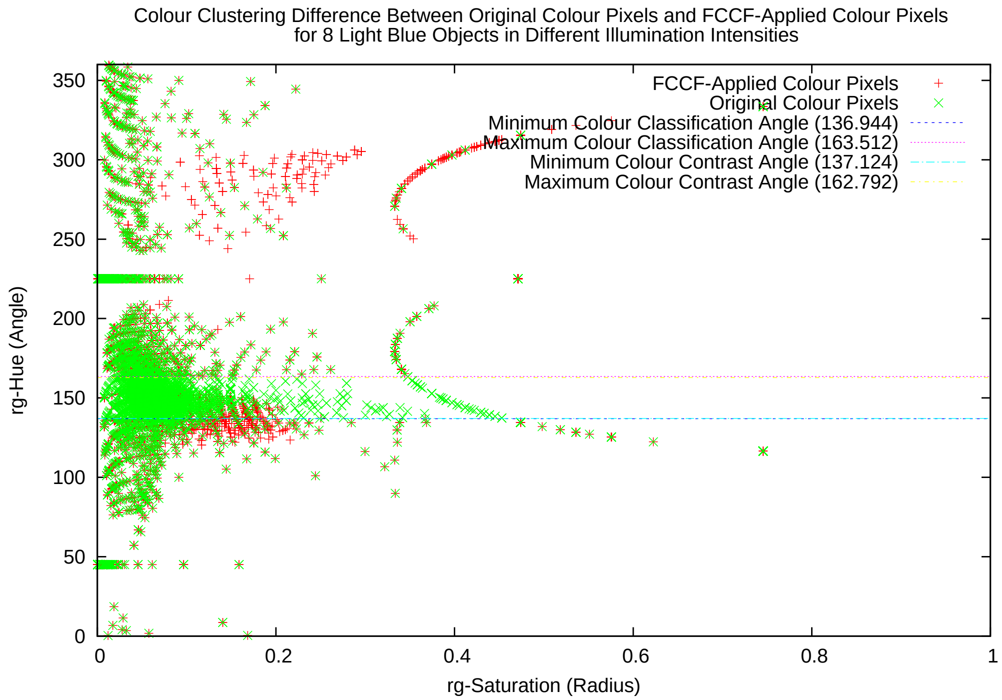

Figure 68: Colour Pixel Clustering on rg-Hue / rg-Saturation Chart for Light Blue Objects

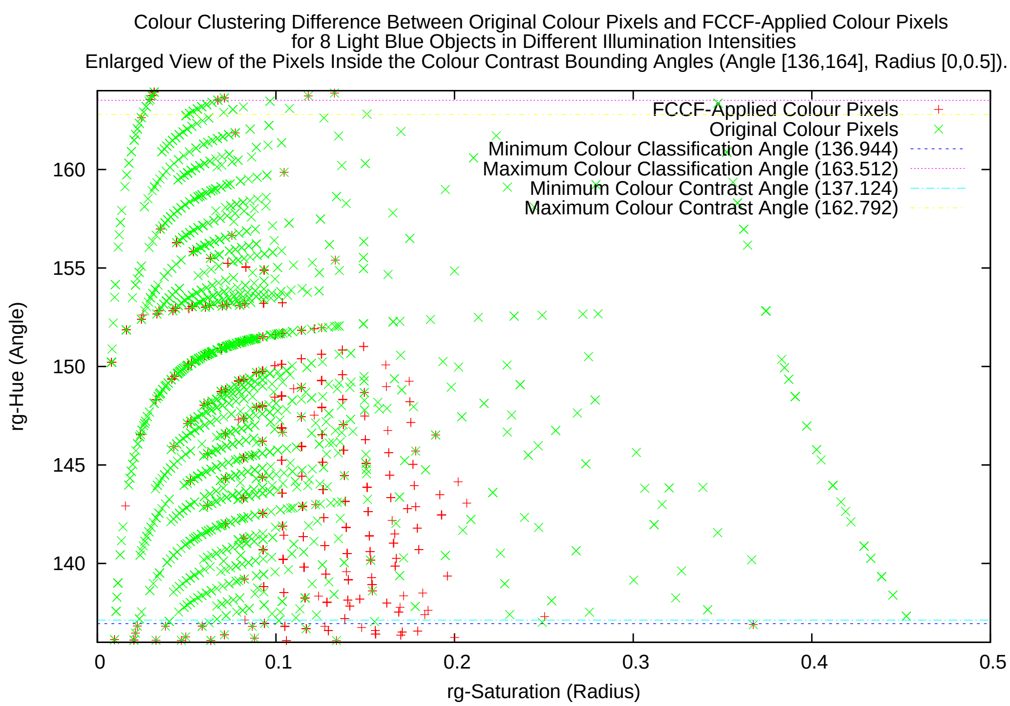

Figure 69: Enlarged Colour Pixel Clustering on rg-Hue / rg-Saturation Chart for Light Blue Objects

5.2.6 Reductions in Memory Usage

As depicted in Table 8, the memory storage requirement for the reduced colour depth LUT optimised for colour classification varies between 16KB and 512KB. Altogether, for the classification of all 6 target colours, the total memory requirement for the Variable Colour Depth LUT is only 984KB. In contrast, a 24-bit colour depth LUT requires 12MB for all 6 colours.

5.2.7 Summary

We tested FCCF with Variable Colour Depth LUT for reducing memory usage and found that FCCF improves colour classification accuracy even when there were lost bits in the colour depth. Automatically optimised FCCF rules suggest that FCCF effectively employs colour contrast rules to compensate for the escaping pixels from the narrow pie-slice decision region for colours like Yellow. These was not known prior to this experiments.

5.3 Fuzzy-Genetic Colour Calibration

Experiments with Different Test-Sets

Each experiment was conducted using two separate test-sets which test on different population sizes and generations. Test results with suffix 'P' mean testing several number of different population sizes on a fixed number of generations. On the other hand, test results with suffix 'G' mean testing several number of different evolution generations on a fixed population size.

Angle Calculation and Range Limiting

The search space for both the pie-slice decision angles and contrast angles ranges from 0 to 360 degrees. However, it is very unlikely that the colour to be classified requires larger than 180 degrees of angle in the pie-slice decision range. The angle range is either supplied by previously discovered solution, or decided upon the extracted minimum and maximum angles taken from the actual colour pixels of the target objects. A tolerance between 30 to 60 degrees is applied to widen the search range, with the limit of less than 180 degrees for the scope of searching.

Guided Search

The search space could be limited to the previously discovered solution. In this experiment, the search space is limited to finding the optimal angles and radii only. The search algorithm is supplied with parameters calibrated previously from Variable Colour Depth with CCRE experiments (i.e. colour contrast rules and VCD parameters). These parameters are no longer included in the search and therefore serves as a 'guide' to the search process. Test results labelled with the suffix 'AO' mean only angles and radii need to be searched for, and these parameters guide the search process.

Standard (Unguided) Search

Using the centre points of all target objects at different illumination conditions, we extract the minimum and maximum bounding angles and radii. Using the same techniques described in the Section 5.3, the bounding search space is adjusted. Test results without suffix 'AO' means that the tests were processed using the Standard (Unguided) search method.

Scaling

Sigma Truncation Scaling is used for scaling function since Sigma Truncation Scaling allows to have objective function to return negative values.

Pcross and Pmutation Rates

The crossover probability (Pcross) and mutation probability (Pmutation) rates are fixed to 0.6 and 0.033 respectively. The optimisation task is then focused on optimising the number of population, generations and search strategy. These numbers were suggested in [27] and used as a general rule of thumb. It is suggested that a choice of a high crossover probability, a low mutation probability (inversely proportional to the population size), and a moderate population size influences good GA performance.

Test Repetition

The Genetic Algorithm test result could end up very poor accuracy score because of inadequate parameters or purely 'unlucky'. When the test score resulted less than 0.5, the test repeated with a different random seed parameter. Test results with suffix 'R' mean that it has repeated the test.

Random Seed

Genetic Algorithm depends on a good random number generator for its Pcross and Pmutation operations. If the random number seed is fixed, the performance of Genetic Algorithms are identical. This influences the effectiveness of Genetic Algorithm parameters such as population size and number of generations. In this experiment, random seed is fixed to 1 for the entire test-sets except during repeated test, where the random seed is set to 2.

Target Image Reduction

In the Genetic Algorithm processing, substantial amount of processing time is spent on evaluating fitness function. In this experiment, the speed of the fitness function depends linearly on the size of the target image. Test results marked with suffix

'H' mean that the tests were processed on reduced target image in 4:1 ratio, which is 25% size of original target image.

5.3.1 Fuzzy-Genetic Colour Calibration Parameters and Scores

Table 10 details parameter configuration of the experiment. The table indicates the experiment name, size of population, number of generations, whether Guided or Standard(Unguided) search is applied and size of target image. Table 11, 12, 13, 14, 15 and 16 details the results of the experiments on Fuzzy-Genetic Colour Calibration for each colour. Name of experiment represents its Genetic Algorithm parameter sets and sorted by best score. The table indicates the performance of the rule combination in terms of the number of hits, misclassifications, parameters of colour classification, colour depth and colour contrast operation used for each of the colour channel, improvement ratio against the best results from table 8 (denoted as 'vLUT op' in Experiment Name). The best results show that Fuzzy-Genetic Colour Calibration improved average accuracy of 9.82% for 6 colours which range from 3.04% to 20.52%.

5.3.2 Colour Contrast Rule Component Distribution

Table 17 shows the distribution of colour contrast rules of Fuzzy-Genetic experiment result. The table excludes results of Fuzzy-Genetic experiment using 'Guidance' strategy because colour contrast rules were not searched and fixed. The table also excludes non-significant results of Fuzzy-Genetic experiment which resulted no improvement against the best results from table 8 (denoted as 'vLUT op' in Experiment Name). Figure 70, 71, 72, 73, 74, and 75 are its corresponding 3D chart, in these charts, the peaks are formed where majority of colour contrast rule components are concentrated and flats where no colour contrast rule component is allocated.

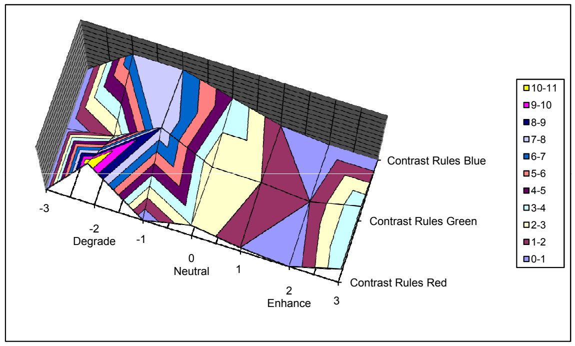

Figure 70: Contrast Rule Component Distribution for Yellow

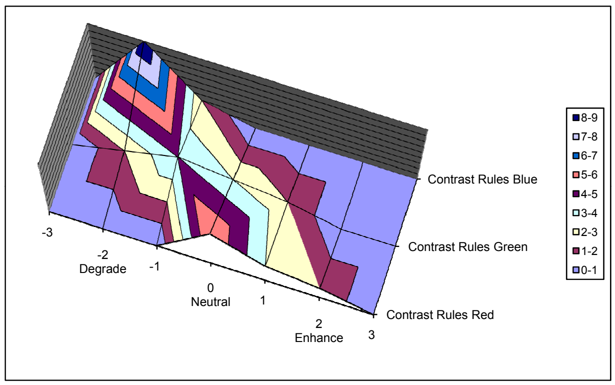

Figure 71: Contrast Rule Component Distribution for Green

5.3.3 Summary

We tested various Genetic Algorithm parameters to find optimised solution for generating accurate colour classification parameters. Obviously larger population and generations generally leads towards better solutions. However, in some colours, much smaller population and generations also resulted to competitive solutions in a fraction of evolutions when processed in parallel with different random seeds. It is interesting that Standard (Unguided) search strategy usually ends up with better result than Guided search strategy suggest that allowing more parameters to explore leads better result than limit search parameters to explore more deeply.

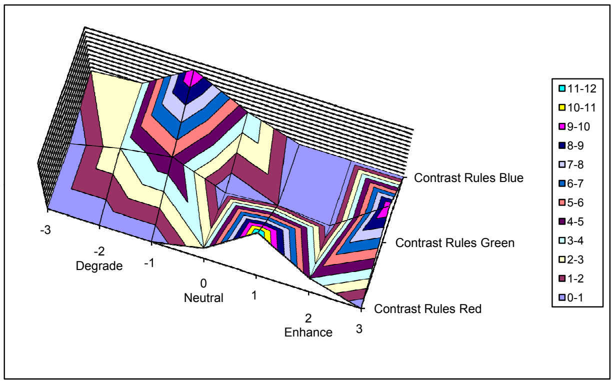

Figure 72: Contrast Rule Component Distribution for Pink

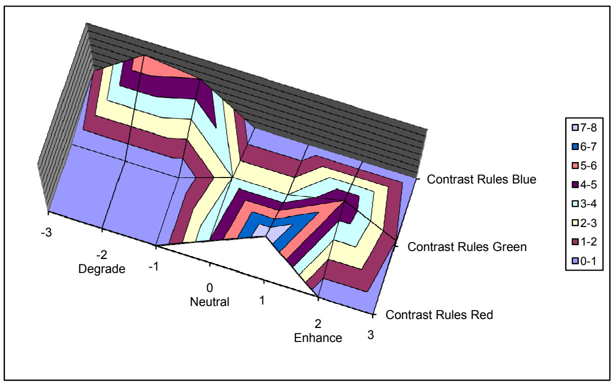

Figure 73: Contrast Rule Component Distribution for Purple

5.4 Discussion

We have extensively tested the effectiveness of Variable Colour Depth and automatic colour classifier calibration using a Fuzzy-Genetic algorithm. Figure 55 indicates that there are observable patterns on colour contrast rules for each target colour and figures 70, 71, 72, 73, 74, and 75 show it more clearly. When there are noticeable peaks and flats, we could say that it developed an optimised pattern to maximise its colour classification accuracy. For example, target colour Purple contrast rule is always enhancing the Red channel, while degrading or keeping the Green and Blue channels fixed.

FCCF modifies the colour attribute of pixels to maximise its classification accuracy. Figures 58 , 60, 62, 64, 66, and 69shows good example of how the FCCF

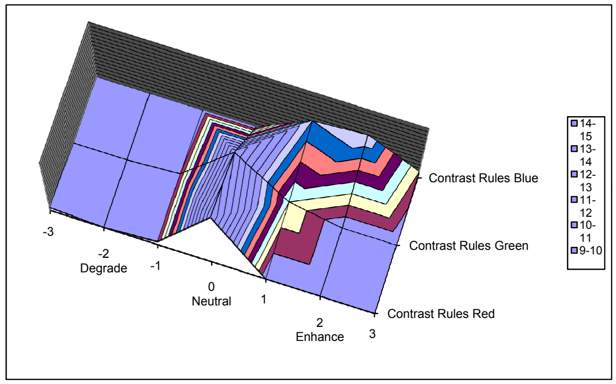

Figure 74: Contrast Rule Component Distribution for Violet

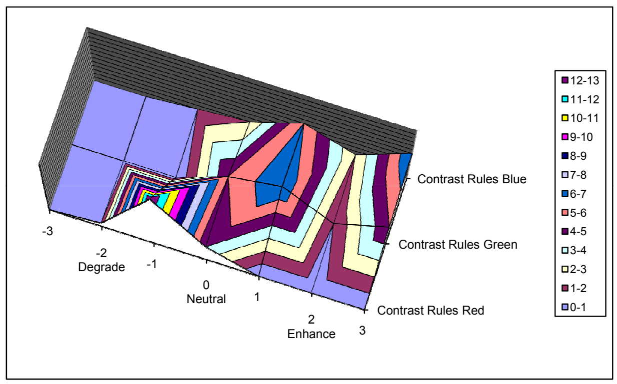

Figure 75: Contrast Rule Component Distribution for Light Blue

improve colour classification adaptively by relocating colour pixels. Table 18 show results of the FCCF process in 6 target colours. For example, in the Yellow target colour, there are total of 4649 target pixels in the entire Yellow objects (0 to 360 degree). Within 4649 pixels, 2372 pixels were located within colour contrast angles (41.832 to 47.808 degree) . From the 2372 pixels, only 56 pixels are found inside the colour classification angles (43.992 to 46.476 degree). After the FCCF processing, 2266 pixels remained within the boundaries of the colour contrast angles and 2205 pixels were moved into colour classification region which classifies the target colour Yellow. On the other hand, in the Green target colour, 976 colour pixels belonging to the colour classification angles were moved out.

The Variable Colour Depth algorithm does not only reduce memory consumption, but also improves colour classification accuracy by influencing the colour pixels position in the colour space. Figure 57 shows how the application of FCCF with reduced colour depth clearly influences the formation of patterns. Most of the colour pixels are lining up with the rg-Hue (angle) axis. This formation help improve the efficacy of of colour contrast rules, as reflected by the scores.

Table 19 shows the difference between manually calibrated colour classification angles and contrast angles, and the Fuzzy-Genetic-optimised colour classification angles and contrast angles. The optimised angles are usually larger than manually calibrated ones because larger contrast angles process more colour pixels within the target area. Likewise, FCCF effectively discards colour pixels that cause misclassifications.

Table 10: Fuzzy-Genetic Colour Calibration Experiment Configuration

Experiment Name	Population	Generation	Guided	Target	Scaling
GA1-P	25	200			
GA2-P	50	200	No	Full	No
GA3-P	100	200			
GA4-P	200	200			
GA1-P-AO	25	200			
GA2-P-AO	50	200	Yes	Full	No
GA3-P-AO	100	200			
GA4-P-AO	200	200			
GA1-P-AO-H	25	200			
GA2-P-AO-H	50	200	Yes	Quarter	No
GA3-P-AO-H	100	200			
GA4-P-AO-H	200	200			
GA1-P-AO-H-STS	25	200			
GA2-P-AO-H-STS	50	200	Yes	Quarter	Sigma Truncation
GA3-P-AO-H-STS	100	200			
GA4-P-AO-H-STS	200	200			
GA1-P-H	25	200			
GA2-P-H	50	200	No	Quarter	No
GA3-P-H	100	200			
GA4-P-H	200	200			
GA1-P-H-STS	25	200			
GA2-P-H-STS	50	200	No	Quarter	Sigma Truncation
GA3-P-H-STS	100	200			
GA4-P-H-STS	200	200			
GA1-P-AO-STS	25	200			
GA2-P-AO-STS	50	200	Yes	Full	Sigma Truncation
GA3-P-AO-STS	100	200			
GA4-P-AO-STS	200	200			
GA1-P-STS	25	200			
GA2-P-STS	50	200	No	Full	Sigma Truncation
GA3-P-STS	100	200			
GA4-P-STS	200	200			
GA1-G	200	25			
GA2-G	200	50	No	Full	No
GA3-G	200	100			
GA1-G-AO	200	25			
GA2-G-AO	200	50	Yes	Full	No
GA3-G-AO	200	100			
GA1-G-AO-H	200	25			
GA2-G-AO-H	200	50	Yes	Quarter	No
GA3-G-AO-H	200	100			
GA1-G-AO-H-STS	200	25			
GA2-G-AO-H-STS	200	50	Yes	Quarter	Sigma Truncation
GA3-G-AO-H-STS	200	100			
GA1-G-H	200	25			
GA2-G-H	200	50	No	Quarter	No
GA3-G-H	200	100			
GA1-G-H-STS	200	25			
GA2-G-H-STS	200	50	No	Quarter	Sigma Truncation
GA3-G-H-STS	200	100			
GA1-G-AO-STS	200	25			
GA2-G-AO-STS	200	50	Yes	Full	Sigma Truncation
GA3-G-AO-STS	200	100			
GA1-G-STS	200	25			
GA2-G-STS	200	50	No	Full	Sigma Truncation
GA3-G-STS	200	100			

Table 11: Fuzzy-Genetic Colour Calibration Result for Yellow

Experiment Name	Score	Hits	Misses	Angle		Radius		Contrast Angle		Contrast Rule	Colour Depth	bits	Running Time	Score
				Min	Max	Min	Max	Min	Max	[R][G][B]	[R][G][B]		(Sec)	Improvement
GA3-P	0.675945	2355	89	32.18	52.42	0.03	0.60	15.70	67.14	[-2][-1][-1]	[7][6][8]	21	1659	103.04%
GA4-P-H	0.673284	2360	95	31.25	52.18	0.03	0.56	15.00	62.16	[-2][-1][-1]	[7][7][8]	22	856	102.64%
GA2-G-H	0.670623	2358	99	16.00	50.95	0.06	0.11	15.76	53.30	[-2][-1][-2]	[7][7][8]	22	224	102.23%
GA3-P-H	0.669324	2314	89	37.99	46.38	0.03	0.61	16.41	47.96	[3][1][-1]	[8][5][8]	21	392	102.03%
GA4-P-H-STS	0.668673	2255	75	30.01	55.88	0.04	0.75	15.76	60.87	[-2][-1][-1]	[8][8][8]	24	783	101.94%
GA4-P	0.668602	2369	105	43.92	52.36	0.00	0.48	15.06	60.16	[-2][-1][1]	[7][6][7]	20	3267	101.92%
GA2-G-AO2	0.664503	2375	114	38.61	45.75	0.05	0.93	19.55	44.78	[1][3][-2]	[7][8][6]	21	829	101.30%
GA1-P-AO-STS	0.664327	2374	114	34.39	45.75	0.05	0.80	19.49	45.39	[1][3][-2]	[7][8][6]	21	404	101.27%
GA3-P-AO	0.664327	2374	114	37.08	45.75	0.05	0.32	19.49	45.26	[1][3][-2]	[7][8][6]	21	1540	101.27%
GA1-P-AO2	0.663998	2372	114	27.55	45.75	0.05	0.55	19.92	44.78	[1][3][-2]	[7][8][6]	21	409	101.22%
GA1-G-AO2	0.663806	2371	114	31.16	45.75	0.05	0.31	19.43	64.20	[1][3][-2]	[7][8][6]	21	403	101.19%
GA2-P-AO2	0.663726	2387	119	19.79	45.75	0.05	0.18	14.85	44.90	[1][3][-2]	[7][8][6]	21	800	101.18%
GA2-P-AO-STS	0.6633	2368	114	27.92	45.75	0.05	0.51	19.86	50.58	[1][3][-2]	[7][8][6]	21	791	101.12%
GA4-P-AO	0.6633	2368	114	27.12	45.75	0.05	0.24	19.92	65.12	[1][3][-2]	[7][8][6]	21	2923	101.12%
GA1-P-AO-H-STS	0.663025	2383	119	27.19	45.75	0.05	0.47	15.34	51.01	[1][3][-2]	[7][8][6]	21	104	101.07%
GA2-P-AO-H-STS	0.663025	2383	119	44.29	45.75	0.05	0.38	14.54	47.34	[1][3][-2]	[7][8][6]	21	211	101.07%
GA3-G-AO-H	0.663025	2383	119	43.07	45.75	0.05	0.51	14.24	51.19	[1][3][-2]	[7][8][6]	21	426	101.07%
GA2-G-AO-H	0.662693	2381	119	44.23	45.75	0.05	0.42	16.01	60.78	[1][3][-2]	[7][8][6]	21	207	101.02%
GA2-P	0.662155	2223	76	43.15	45.97	0.06	0.93	20.04	47.67	[3][0][-2]	[7][7][8]	22	724	100.94%
GA3-P-STS	0.662018	2357	113	42.74	46.79	0.02	0.86	22.68	58.52	[1][3][0]	[7][8][6]	21	1599	100.92%
GA3-G-H	0.661953	2285	93	19.34	47.96	0.02	0.41	15.65	68.84	[0][3][0]	[7][6][8]	21	420	100.91%
GA2-G-AO-STS	0.661615	2358	114	39.16	45.75	0.05	0.67	21.93	54.12	[1][3][-2]	[7][8][6]	21	855	100.86%
GA4-P-STS	0.661466	2234	80	30.25	46.67	0.03	0.11	25.62	53.36	[0][-2][-1]	[8][6][8]	22	3187	100.84%
GA1-G2	0.661139	2236	81	40.22	49.25	0.00	0.75	37.46	55.70	[-2][3][-2]	[8][6][8]	22	432	100.79%
GA1-G-H-STS	0.660896	2219	77	43.92	52.01	0.03	0.89	40.63	52.18	[-2][3][-1]	[8][8][8]	24	109	100.75%
GA3-P-AO-H-STS	0.660459	2375	121	21.02	45.81	0.06	0.31	19.12	47.40	[1][3][-2]	[7][8][6]	21	415	100.68%
GA1-P	0.660137	2222	79	42.92	45.56	0.02	0.96	26.50	67.67	[-2][2][0]	[7][6][8]	21	367	100.63%
GA1-P-STS2	0.659839	2331	109	41.57	49.84	0.05	0.27	20.69	65.79	[-2][-1][-2]	[8][7][6]	21	386	100.59%
GA1-P-H2	0.659727	2227	81	43.21	45.26	0.06	0.92	33.65	50.19	[3][-1][-2]	[8][6][8]	22	130	100.57%
GA3-G-AO-STS2	0.659562	2287	97	38.00	45.20	0.05	0.91	36.96	62.00	[1][3][-2]	[7][8][6]	21	1807	100.55%
GA3-P-AO-STS	0.659488	2301	101	37.87	45.08	0.05	0.91	32.62	61.76	[1][3][-2]	[7][8][6]	21	1560	100.53%
GA4-P-AO-H-STS	0.65941	2286	97	40.74	45.69	0.05	0.36	37.75	69.02	[1][3][-2]	[7][8][6]	21	837	100.52%
GA4-P-AO-STS	0.65941	2286	97	44.90	45.69	0.05	0.18	37.75	55.95	[1][3][-2]	[7][8][6]	21	3213	100.52%
GA4-P-AO-H	0.659405	2286	97	42.21	45.33	0.05	0.57	37.14	51.37	[1][3][-2]	[7][8][6]	21	802	100.52%
GA2-G	0.658747	2300	102	30.95	46.91	0.03	0.59	16.70	53.42	[0][-2][-1]	[8][6][7]	21	745	100.42%
GA2-P-H-STS	0.658608	2243	87	45.56	49.25	0.02	0.96	15.47	49.31	[-2][0][-2]	[8][5][8]	21	202	100.40%
GA2-P-AO-H	0.658259	2290	100	37.51	45.33	0.05	0.94	37.63	72.20	[1][3][-2]	[7][8][6]	21	199	100.35%
GA3-P-AO-H	0.657897	2284	99	44.65	46.24	0.06	0.82	37.87	66.95	[1][3][-2]	[7][8][6]	21	391	100.29%
GA3-G-AO-H-STS	0.657868	2319	109	32.25	46.18	0.06	0.96	29.81	46.85	[1][3][-2]	[7][8][6]	21	449	100.29%
GA3-P-H-STS	0.657846	2253	91	28.90	51.89	0.06	0.40	23.21	54.41	[-2][1][-2]	[8][8][8]	24	399	100.28%
GA1-P-AO-H	0.657441	2303	105	29.93	45.63	0.05	0.41	29.38	47.77	[1][3][-2]	[7][8][6]	21	100	100.22%
GA2-P-H	0.657118	2195	77	38.87	44.21	0.01	0.82	20.75	47.79	[3][-1][0]	[8][8][8]	24	195	100.17%
GA1-G-AO-H2	0.656917	2296	104	44.35	46.55	0.05	0.94	37.94	48.75	[1][3][-2]	[7][8][6]	21	104	100.14%
vLUT op	0.655979	2261	96	43.99	46.48	0.00	1.00	41.83	47.81	[1][3][-2]	[7][8][6]	21		100.00%
GA2-G-STS	0.655953	2363	125	20.40	54.71	0.02	0.64	17.93	72.07	[-1][2][0]	[8][5][6]	19	741	100.00%
GA2-G-AO-H-STS	0.655337	2315	112	29.08	45.33	0.05	0.76	21.14	52.96	[1][3][-2]	[7][8][6]	21	225	99.90%
GA3-G-AO	0.65516	2314	112	36.96	45.14	0.05	0.72	21.44	73.79	[1][3][-2]	[7][8][6]	21	1572	99.88%
GA2-P-STS	0.654361	2255	97	46.03	54.00	0.05	0.65	45.50	69.08	[-2][-1][-2]	[8][7][6]	21	777	99.75%
GA1-P-H-STS	0.653969	2338	121	32.13	55.82	0.02	0.23	20.04	58.81	[-1][3][0]	[8][6][6]	20	99	99.69%
GA3-G-H-STS	0.652612	2157	75	20.81	66.03	0.02	0.65	19.16	73.94	[-2][3][0]	[8][6][7]	21	431	99.49%
GA3-G-STS	0.652436	2298	112	27.14	46.50	0.05	0.98	19.40	50.78	[0][-2][-2]	[8][7][6]	21	1495	99.46%
GA1-G-H	0.651883	2113	65	25.73	43.80	0.06	0.26	15.59	46.91	[3][-2][-2]	[8][8][7]	23	106	99.38%
Reference	0.64853	2104	68	43.99	46.48	0.00	1.00	41.83	47.81	[3][1][-2]	[8][8][8]	24		98.86%
GA3-G	0.634787	1992	62	20.69	50.54	0.06	0.13	16.94	68.37	[-2][-1][-2]	[7][7][8]	22	1554	96.77%
GA2-G-H-STS	0.622755	1897	57	43.97	45.38	0.04	0.32	20.75	45.91	[-1][3][-1]	[7][6][8]	21	218	94.94%
GA1-G-AO-H-STS	0.618622	1857	54	42.45	45.39	0.06	0.69	40.38	44.71	[1][3][-2]	[7][8][6]	21	114	94.31%
GA1-G-AO-STS2	0.617803	2354	188	44.41	46.12	0.04	0.67	41.23	44.84	[1][3][-2]	[7][8][6]	21	556	94.18%
GA1-G-STS	0.562476	2386	298	34.77	52.07	0.02	0.34	32.24	60.81	[0][2][0]	[6][8][7]	21	377	85.75%
GA1-G-AO-STS	0.416841	2587	674	29.81	49.60	0.05	0.77	15.34	48.26	[1][3][-2]	[7][8][6]	21	430	63.54%
GA3-G-AO-STS	0.397085	480	1	26.76	46.91	0.09	0.51	14.05	45.08	[1][3][-2]	[7][8][6]	21	1648	60.53%
GA1-G-AO-H	0.374017	536	10	39.22	48.56	0.09	0.69	14.11	48.26	[1][3][-2]	[7][8][6]	21	107	57.02%
GA1-G	0.371003	48	1	15.88	48.55	0.18	0.49	15.47	67.79	[0][-2][2]	[6][8][7]	21	410	56.56%
GA1-G-AO	0.369082	90	2	15.09	58.64	0.09	0.79	13.99	61.51	[1][3][-2]	[7][8][6]	21	417	56.26%
GA1-P-STS	0.346724	26	1	57.05	74.00	0.15	0.26	30.89	74.88	[-2][0][2]	[8][5][7]	20	386	52.86%
GA2-G-AO	0.270179	1588	151	43.25	47.71	0.07	0.99	24.44	46.67	[1][3][-2]	[7][8][6]	21	794	41.19%
GA1-P-H	0.219608	60	8	21.86	33.71	0.14	0.70	21.33	43.74	[0][-1][3]	[6][7][5]	18	96	33.48%
GA1-P-AO	0	0	0	28.90	28.90	0.45	0.96	35.19	54.98	[1][3][-2]	[7][8][6]	21	366	0.00%
GA2-P-AO	0	0	0	33.23	38.48	0.26	0.54	18.57	25.35	[1][3][-2]	[7][8][6]	21	761	0.00%

Table 12: Fuzzy-Genetic Colour Calibration Result for Green

Experiment Name	Score	Hits	Misses	Angle Min	Angle Max	Radius Min	Radius Max	Contrast Angle Min	Contrast Angle Max	Contrast Rule [R][G][B]	Colour Depth [R][G][B]	bits	Running Time (Sec)	Score Improvement
GA2-P-STS2	0.695366	3637	128	44.53	101.23	0.04	0.12	37.95	121.46	[0][-1][-2]	[8][8][5]	21	842	108.81%
GA4-P2	0.687847	3424	90	47.01	102.66	0.05	0.86	45.68	123.56	[1][0][-2]	[6][6][7]	19	3792	107.63%
GA2-G-AO	0.683699	3407	105	47.07	100.11	0.06	0.50	26.70	127.52	[1][0][-2]	[6][5][8]	19	1031	106.99%
GA2-P-H-STS	0.683177	3451	111	44.72	94.45	0.03	0.09	40.33	119.27	[0][-1][-1]	[6][8][8]	22	212	106.90%
GA1-P-AO-STS	0.683058	3412	108	47.67	99.06	0.06	0.85	36.44	128.27	[1][0][-2]	[6][5][8]	19	466	106.88%
GA4-P-AO-H-STS	0.682847	3408	107	46.03	103.55	0.06	0.46	35.69	126.77	[1][0][-2]	[6][5][8]	19	935	106.85%
GA2-G-AO-H	0.682642	3401	105	47.37	99.36	0.06	0.64	33.29	126.17	[1][0][-2]	[6][5][8]	19	238	106.82%
GA2-P-AO-STS	0.682563	3391	102	47.22	100.11	0.06	0.73	33.29	123.93	[1][0][-2]	[6][5][8]	19	892	106.81%
GA1-P-AO-H-STS	0.682559	3400	105	47.37	104.60	0.06	0.42	35.99	126.17	[1][0][-2]	[6][5][8]	19	118	106.81%
GA3-G-AO-H-STS	0.682327	3398	105	46.92	97.86	0.06	0.32	46.03	126.17	[1][0][-2]	[6][5][8]	19	508	106.77%
GA4-P-AO-STS	0.682224	3415	111	47.67	101.30	0.06	0.89	37.64	130.22	[1][0][-2]	[6][5][8]	19	3633	106.75%
GA2-P-AO-H	0.682053	3403	107	45.88	104.30	0.06	0.61	34.49	126.17	[1][0][-2]	[6][5][8]	19	267	106.73%
GA4-P-AO-H	0.682053	3403	107	46.18	104.45	0.06	0.92	38.98	126.17	[1][0][-2]	[6][5][8]	19	1051	106.73%
GA2-P-AO-H-STS	0.681792	3393	105	46.47	90.22	0.06	0.55	21.91	126.17	[1][0][-2]	[6][5][8]	19	243	106.69%
GA3-G-AO-H2	0.681047	3410	112	46.47	106.25	0.06	0.50	29.85	126.77	[1][0][-2]	[6][5][8]	19	530	106.57%
GA1-P2	0.68092	3316	72	45.49	66.58	0.05	0.49	44.34	106.29	[1][0][-2]	[7][8][8]	23	444	106.55%
GA1-G-AO	0.680531	3376	104	47.37	79.28	0.06	0.58	37.49	128.72	[1][0][-2]	[6][5][8]	19	529	106.49%
GA1-G-STS2	0.679443	3334	84	47.68	71.54	0.05	0.81	41.38	114.30	[1][0][-2]	[6][6][7]	19	417	106.32%
GA3-G-AO-STS	0.678554	3488	135	46.33	97.11	0.05	0.23	42.73	124.52	[1][0][-2]	[6][5][8]	19	1858	106.18%
GA4-P-H2	0.678445	3318	86	50.55	90.82	0.03	0.29	49.78	118.22	[0][-1][-1]	[8][7][8]	23	1022	106.16%
GA4-P-STS	0.674115	3226	70	43.29	92.45	0.05	0.25	35.47	122.61	[0][-2][-2]	[8][8][6]	22	3423	105.49%
GA2-G-AO-STS	0.673427	3469	143	47.07	77.34	0.05	0.57	33.29	132.46	[1][0][-2]	[6][5][8]	19	969	105.38%
GA3-P-AO-H2	0.671607	3207	90	46.03	106.25	0.06	0.46	36.74	128.87	[1][0][-2]	[6][5][8]	19	442	105.09%
GA3-P-AO-H-STS	0.670509	3343	113	47.37	66.25	0.06	0.28	30.75	118.38	[1][0][-2]	[6][5][8]	19	471	104.92%
GA4-P-H-STS	0.670414	3413	139	44.82	103.04	0.06	0.45	43.77	118.69	[2][1][-2]	[8][8][8]	24	851	104.91%
GA2-P-AO	0.668266	3152	82	46.33	70.59	0.06	0.35	19.96	125.57	[1][0][-2]	[6][5][8]	19	844	104.57%
GA2-G-AO-H-STS	0.667925	3359	122	45.58	67.15	0.05	0.32	22.06	111.64	[1][0][-2]	[6][5][8]	19	254	104.52%
GA3-P-AO-STS	0.666283	3131	81	45.28	70.14	0.06	0.71	35.99	124.22	[1][0][-2]	[6][5][8]	19	1791	104.26%
GA1-P-H-STS	0.655742	3264	135	45.39	109.82	0.06	0.10	50.93	123.75	[0][-1][-2]	[8][8][6]	22	117	102.61%
GA3-P-STS2	0.654836	3183	107	46.92	59.90	0.06	0.31	41.67	123.08	[2][1][-2]	[6][5][7]	18	1693	102.47%
GA1-G-AO-STS	0.652177	3305	142	47.67	78.83	0.05	0.49	16.06	101.90	[1][0][-2]	[6][5][8]	19	481	102.05%
GA2-G-H-STS2	0.641004	3041	101	49.21	91.11	0.04	0.76	32.98	116.59	[0][-2][-2]	[8][6][6]	20	288	100.30%
vLUT op	0.639059	3137	127	45.58	96.66	0.05	1.00	45.29	96.06	[1][0][-2]	[6][5][8]	19		100.00%
GA2-G-STS	0.636304	3093	117	48.35	101.42	0.01	0.97	28.12	123.08	[1][-1][-1]	[6][7][8]	21	821	99.57%
GA3-G	0.635923	3127	138	52.07	87.29	0.02	0.76	50.26	111.54	[0][-1][-1]	[8][7][7]	22	1587	99.51%
GA4-P-AO	0.596873	3488	362	35.24	81.83	0.06	0.40	45.73	137.71	[1][0][-2]	[6][5][8]	19	4120	93.40%
Reference	0.552422	3313	383	45.58	96.66	0.05	1.00	45.29	96.06	[0][-1][-3]	[8][8][8]	24		86.44%
GA3-G-STS2	0.551099	3234	361	45.58	75.84	0.05	0.73	43.86	122.13	[0][3][-2]	[6][8][8]	22	1662	86.24%
GA3-G-AO	0.532837	3388	468	45.88	68.05	0.01	0.37	1.53	102.50	[1][0][-2]	[6][5][8]	19	1641	83.38%
GA3-G-H-STS	0.518213	2586	247	51.50	83.47	0.02	0.60	51.12	96.55	[0][2][-1]	[5][7][5]	17	464	81.09%
GA3-P-AO	0.507483	3678	668	47.37	102.05	0.02	0.73	28.50	128.27	[1][0][-2]	[6][5][8]	19	1566	79.41%
GA3-G-STS	0.398293	820	1	61.43	111.92	0.03	0.21	57.04	115.54	[0][-1][-3]	[8][8][8]	24	1623	62.32%
GA3-P	0.397085	480	1	95.12	114.59	0.14	0.47	73.45	115.54	[-3][1][-2]	[5][7][5]	17	1784	62.14%
GA2-P2	0.395531	313	1	88.25	104.38	0.11	0.23	66.77	105.33	[-1][0][-2]	[8][5][6]	19	889	61.89%
GA2-G2	0.395459	308	1	78.22	89.58	0.03	0.49	77.37	112.39	[0][3][0]	[5][7][5]	17	869	61.88%
GA3-P-STS	0.393921	230	1	65.91	74.22	0.00	0.64	59.90	119.84	[2][-2][0]	[5][7][7]	19	1706	61.64%
GA1-G-AO-H-STS2	0.393421	425	2	51.27	60.11	0.06	0.94	11.72	101.00	[1][0][-2]	[6][5][8]	19	162	61.56%
GA4-P	0.393079	202	1	79.66	92.26	0.03	0.94	76.51	110.77	[0][-2][-1]	[7][5][5]	17	3396	61.51%
GA1-P	0.389087	128	1	114.59	116.50	0.01	0.06	42.05	100.85	[0][0][1]	[6][7][5]	18	386	60.88%
GA1-G-H-STS2	0.386687	839	8	45.11	94.07	0.08	0.88	31.46	123.47	[3][0][-2]	[5][5][6]	16	146	60.51%
GA3-P-H-STS	0.384948	1020	11	61.43	89.58	0.06	0.23	36.71	121.37	[0][-1][-2]	[8][6][6]	20	422	60.24%
GA2-P-STS	0.379192	134	2	80.32	83.00	0.02	0.96	60.47	101.80	[-1][3][0]	[7][7][6]	20	828	59.34%
GA2-P	0.378878	66	1	42.15	115.64	0.34	0.81	87.48	118.12	[1][2][2]	[8][7][7]	22	830	59.29%
GA3-G-H2	0.377877	504	8	113.92	119.65	0.07	0.44	79.27	120.60	[-2][-1][-1]	[8][6][6]	20	458	59.13%
GA1-G-STS	0.377155	61	1	91.87	97.89	0.10	0.33	72.02	114.78	[0][1][-2]	[6][5][8]	19	402	59.02%
GA3-P-H2	0.376881	663	11	85.19	109.34	0.12	0.30	68.78	109.24	[-2][-1][-2]	[7][7][5]	19	560	58.97%
GA3-G-H	0.376384	708	12	70.02	80.04	0.04	0.50	60.57	85.76	[-2][0][-2]	[8][8][5]	21	461	58.90%
GA1-G-AO-H-STS	0.373324	522	10	52.32	66.25	0.04	0.51	33.29	119.58	[1][0][-2]	[6][5][8]	19	126	58.42%
GA4-P-H	0.372122	799	16	75.08	121.08	0.13	0.19	75.55	123.47	[-2][0][-2]	[7][7][5]	19	1195	58.23%
GA1-P-STS	0.371592	49	1	102.66	113.64	0.15	0.25	34.70	122.51	[-1][1][-1]	[8][8][5]	21	421	58.15%
GA2-G-H-STS	0.371003	192	4	61.33	78.32	0.07	0.34	55.51	83.38	[-1][1][-1]	[5][5][8]	18	237	58.05%
GA3-P2	0.360317	35	1	59.14	117.36	0.31	0.78	49.40	119.36	[0][2][3]	[6][7][6]	19	1778	56.38%
GA1-G-H2	0.355152	2475	80	50.64	69.63	0.05	0.95	28.78	110.96	[1][-1][-2]	[5][5][8]	18	111	55.57%
GA1-G-AO-H2	0.355062	494	16	52.02	61.01	0.03	0.45	15.76	126.77	[1][0][-2]	[6][5][8]	19	134	55.56%
GA2-G-H2	0.351563	229	8	78.99	83.09	0.04	0.17	63.53	112.97	[-2][0][-2]	[8][8][5]	21	240	55.01%
GA1-P-STS2	0.350493	28	1	104.28	117.83	0.50	0.68	85.00	111.06	[2][3][3]	[5][5][7]	17	400	54.85%
GA3-P-H-STS2	0.329798	3127	159	46.06	79.27	0.02	0.52	34.03	120.51	[2][0][-2]	[6][6][8]	20	575	51.61%
GA1-G-H-STS	0.318834	662	39	61.24	98.17	0.06	0.10	50.35	108.10	[-2][2][-1]	[8][8][5]	21	120	49.89%
GA1-G2	0.309825	1571	103	57.99	76.60	0.02	0.31	53.79	84.91	[0][2][0]	[8][8][5]	21	451	48.48%
GA1-G	0.306286	44	3	85.57	99.22	0.04	0.90	84.91	106.19	[-2][2][-2]	[8][8][8]	24	442	47.93%
GA1-G-H	0.287529	1644	135	45.58	57.04	0.06	0.73	28.59	101.04	[0][2][-2]	[6][7][7]	20	98	44.99%
GA2-G	0.275758	11	1	95.79	114.59	0.61	0.89	36.52	118.69	[1][1][2]	[8][5][8]	21	799	43.15%
GA1-P-H2	0.216646	59	8	102.47	116.50	0.06	0.89	96.26	114.97	[1][0][0]	[7][7][6]	20	140	33.90%
GA2-G-H	0.057602	1947	506	57.23	92.06	0.01	0.75	31.46	93.78	[1][0][0]	[5][5][8]	18	210	9.01%
GA1-G-AO-H	0	0	0	26.70	117.18	0.64	0.74	1.98	117.78	[1][0][-2]	[6][5][8]	19	111	0.00%
GA1-P-AO	0	0	0	87.07	95.16	0.26	0.79	87.22	136.36	[1][0][-2]	[6][5][8]	19	399	0.00%
GA1-P-AO2	0	0	0	26.25	48.72	0.33	0.75	61.90	90.52	[1][0][-2]	[6][5][8]	19	412	0.00%
GA1-P-AO-H	0	0	0	10.22	357.64	0.81	0.82	25.65	79.13	[1][0][-2]	[6][5][8]	19	103	0.00%
GA1-P-AO-H2	0	0	0	46.47	77.34	0.35	0.95	89.77	99.21	[1][0][-2]	[6][5][8]	19	106	0.00%
GA3-G-AO-H	0	0	1	66.10	81.38	0.27	0.58	16.21	70.14	[1][0][-2]	[6][5][8]	19	418	0.00%
GA1-P-H	-0.11429	5	2	71.07	74.22	0.12	0.83	33.37	108.00	[-2][3][0]	[8][5][6]	19	106	-17.88%
GA3-P-AO-H	-0.39192	22	14	113.89	134.56	0.11	0.29	21.01	136.81	[1][0][-2]	[6][5][8]	19	422	-61.33%
GA3-P-H	-0.40338	82	53	67.82	119.17	0.19	0.59	31.36	122.89	[2][2][1]	[5][7][7]	19	438	-63.12%
GA2-P-H2	-0.6268	32	27	105.24	121.08	0.16	0.59	29.07	121.18	[2][1][0]	[8][8][5]	21	288	-98.08%
GA2-P-H	-1.01818	30	36	61.33	81.76	0.15	0.26	50.16	96.26	[-2][0][3]	[7][7][6]	20	213	-159.32%

Table 13: Fuzzy-Genetic Colour Calibration Result for Pink

Experiment Name	Score	Hits	Misses	Angle		Radius		Contrast Angle		Contrast Rule	Colour Depth	bits	Running Time	Score
				Min	Max	Min	Max	Min	Max	[R][G][B]	[R][G][B]		(Sec)	Improvement
GA3-G2	0.705861	2223	50	283.44	335.43	0.16	0.65	280.51	335.37	[2][-2][0]	[8][8][7]	23	1866	113.34%
GA1-P	0.702445	2073	21	306.63	335.43	0.16	0.97	292.23	338.30	[1][-1][-3]	[8][8][5]	21	433	112.79%
GA4-P-STS	0.698391	2175	50	309.00	322.47	0.19	0.51	297.22	325.27	[1][3][-1]	[7][5][7]	19	3374	112.14%
GA3-G-H	0.69794	2058	26	308.44	328.08	0.29	0.74	275.02	329.39	[1][3][-1]	[6][5][7]	18	462	112.07%
GA1-G2	0.697039	2070	30	308.44	322.15	0.28	0.97	282.94	329.01	[1][3][-1]	[6][5][7]	18	491	111.93%
GA3-P-H2	0.695599	2249	72	284.12	332.44	0.16	0.85	285.12	331.38	[2][-2][0]	[8][6][6]	20	603	111.69%
GA4-P-H-STS	0.695028	2244	76	308.31	337.30	0.15	0.30	303.01	337.55	[1][-1][-2]	[6][8][5]	19	830	111.60%
GA2-P	0.693808	2161	53	305.76	324.96	0.15	0.71	294.60	326.27	[0][3][-2]	[6][5][6]	17	793	111.41%
GA2-P-H-STS	0.691939	2235	78	308.56	328.26	0.18	0.79	295.91	331.69	[1][3][-1]	[8][8][8]	24	214	111.11%
GA3-P	0.687431	1981	22	306.07	329.64	0.26	0.68	277.58	330.88	[0][3][-3]	[7][5][5]	17	1877	110.38%
GA1-G-H2	0.686431	2042	39	308.69	324.21	0.29	0.83	278.70	333.69	[1][3][-1]	[6][5][8]	19	127	110.22%
GA1-P-H-STS	0.68107	2301	117	304.45	333.75	0.14	0.27	293.66	337.55	[1][-1][-1]	[6][8][7]	21	105	109.36%
GA2-P-H	0.68041	2013	41	312.12	335.06	0.31	0.47	293.54	335.68	[2][3][-1]	[8][5][7]	20	268	109.25%
GA3-G-STS	0.676062	2002	44	312.49	322.28	0.30	0.69	281.75	334.25	[2][3][-2]	[7][6][8]	21	1640	108.56%
GA3-P-AO	0.663689	2127	97	311.77	338.12	0.14	0.30	302.80	356.92	[1][-1][0]	[7][8][7]	22	1703	106.57%
GA2-P-AO	0.660601	2138	105	311.34	339.54	0.14	0.25	293.47	355.50	[1][-1][0]	[7][8][7]	22	751	106.07%
GA3-P-AO-STS	0.660336	2059	83	312.41	337.98	0.14	0.25	304.58	349.09	[1][-1][0]	[7][8][7]	22	1674	106.03%
GA3-G-AO-STS	0.659884	2151	110	311.13	339.62	0.14	0.28	290.55	341.68	[1][-1][0]	[7][8][7]	22	1736	105.96%
GA2-P-AO-H-STS	0.658804	2159	114	310.70	339.97	0.14	0.30	293.97	356.64	[1][-1][0]	[7][8][7]	22	222	105.79%
GA4-P-AO	0.658563	2067	88	311.77	337.84	0.14	0.25	303.79	354.64	[1][-1][0]	[7][8][7]	22	3374	105.75%
GA2-P-AO-H	0.658516	2078	91	312.41	351.72	0.14	0.25	304.08	355.99	[1][-1][0]	[7][8][7]	22	228	105.74%
GA3-G-AO-H	0.65782	2164	117	310.63	355.78	0.14	0.29	290.41	356.92	[1][-1][0]	[7][8][7]	22	472	105.63%
GA3-G-AO-H-STS	0.657069	2089	97	312.48	333.49	0.14	0.28	293.61	349.66	[1][-1][0]	[7][8][7]	22	480	105.51%
GA2-G-AO	0.657053	2016	78	311.77	339.83	0.14	0.30	300.38	348.66	[1][-1][0]	[7][8][7]	22	910	105.50%
GA1-G-AO-H2	0.65683	2089	97	312.41	337.91	0.14	0.25	302.30	348.66	[1][-1][0]	[7][8][7]	22	119	105.47%
GA3-P-AO-H	0.656502	2086	97	312.48	334.13	0.14	0.26	284.99	348.37	[1][-1][0]	[7][8][7]	22	402	105.42%
GA4-P-AO-STS	0.655769	2017	80	311.41	339.40	0.14	0.25	295.18	355.78	[1][-1][0]	[7][8][7]	22	3520	105.30%
GA1-P-AO-H-STS	0.655087	2094	101	312.41	351.79	0.14	0.24	298.67	351.79	[1][-1][0]	[7][8][7]	22	112	105.19%
GA1-P-AO-H	0.655032	2090	100	311.56	347.66	0.14	0.25	288.98	349.51	[1][-1][0]	[7][8][7]	22	107	105.18%
GA1-P-AO-STS	0.651229	2096	108	311.13	327.58	0.14	0.69	302.16	327.72	[1][-1][0]	[7][8][7]	22	432	104.57%
GA4-P-AO-H2	0.651129	2108	111	310.63	355.35	0.14	0.24	286.13	357.13	[1][-1][0]	[7][8][7]	22	981	104.55%
GA2-G-H-STS	0.650155	2227	146	308.19	328.58	0.20	0.95	291.98	337.74	[1][3][-1]	[5][8][8]	21	232	104.40%
GA1-G-AO-H-STS	0.642991	2068	113	308.28	346.52	0.14	0.29	285.35	353.93	[1][-1][0]	[7][8][7]	22	119	103.25%
GA3-P-H-STS	0.640432	1939	72	298.65	333.56	0.13	0.57	277.76	333.25	[1][-2][0]	[7][8][5]	20	418	102.84%
GA2-P-AO-STS	0.638757	2187	154	304.86	351.44	0.14	0.24	286.85	354.78	[1][-1][0]	[7][8][7]	22	840	102.57%
GA1-G-STS	0.635931	1854	64	319.79	330.45	0.14	0.81	278.51	336.68	[1][-1][-1]	[5][7][5]	17	399	102.11%
GA3-G-H-STS	0.635553	2005	107	304.07	330.26	0.14	0.36	299.96	332.88	[1][-1][0]	[5][8][6]	19	445	102.05%
GA2-G-AO-STS	0.633678	1798	58	311.27	351.65	0.15	0.24	284.99	352.79	[1][-1][0]	[7][8][7]	22	905	101.75%
GA1-G-AO-STS	0.63151	2152	156	304.36	330.07	0.14	0.34	297.03	330.93	[1][-1][0]	[7][8][7]	22	453	101.40%
GA1-P-H	0.630143	1945	95	306.82	329.82	0.14	0.38	305.76	332.82	[0][1][-1]	[7][8][6]	21	102	101.18%
vLUT op	0.622773	1679	46	314.42	327.28	0.15	1.00	275.26	331.88	[1][-1][0]	[7][8][7]	22		100.00%
GA2-G-AO-H-STS	0.614933	2285	227	311.13	351.08	0.13	0.32	301.66	356.92	[1][-1][0]	[7][8][7]	22	240	98.74%
GA3-P-AO-H-STS	0.613629	2195	199	310.63	342.18	0.14	0.51	295.04	343.18	[1][-1][0]	[7][8][7]	22	438	98.53%
GA4-P-AO-H-STS	0.608958	2239	222	310.49	338.05	0.14	0.85	298.10	346.59	[1][-1][0]	[7][8][7]	22	864	97.78%
GA1-G-AO	0.597344	1713	86	315.76	329.50	0.14	0.41	301.80	331.43	[1][-1][0]	[7][8][7]	22	405	95.92%
Reference	0.586446	1714	99	314.42	327.28	0.15	1.00	275.26	331.88	[1][-1][0]	[8][8][8]	24		94.17%
GA2-P-STS2	0.563632	1670	111	291.73	322.72	0.11	0.28	278.82	335.50	[0][-1][1]	[5][8][6]	19	830	90.50%
GA3-G-AO	0.555001	2256	318	300.87	325.44	0.13	0.44	291.76	330.71	[1][-1][0]	[7][8][7]	22	1588	89.12%
GA3-P-STS	0.535381	2098	279	280.20	335.25	0.17	0.22	285.50	329.07	[2][-2][-2]	[7][8][6]	21	1684	85.97%
GA2-G-STS	0.527677	2119	310	309.00	327.70	0.16	0.77	293.72	332.69	[1][1][0]	[5][5][7]	17	818	84.73%
GA1-G-H-STS2	0.51272	1797	214	308.13	325.33	0.09	0.72	307.75	328.76	[1][3][-1]	[7][5][6]	18	140	82.33%
GA1-G-AO-H	0.409178	2160	528	308.57	356.56	0.14	0.73	304.86	354.71	[1][-1][0]	[7][8][7]	22	103	65.70%
GA4-P	0.398675	1056	1	296.22	305.45	0.09	0.97	291.98	314.17	[0][-2][1]	[8][6][7]	21	3330	64.02%
GA2-P-STS	0.398541	959	1	282.63	317.79	0.13	0.57	279.88	321.41	[0][-1][3]	[7][7][8]	22	827	63.99%
GA1-P-STS2	0.397644	594	1	302.70	310.50	0.08	0.12	285.50	331.19	[-1][-2][0]	[8][6][8]	22	425	63.85%
GA1-P-AO2	0.39644	393	1	320.89	329.79	0.16	0.93	286.13	330.78	[1][-1][0]	[7][8][7]	22	419	63.66%
GA4-P-H	0.396252	1493	4	309.56	329.82	0.05	0.43	279.45	331.26	[-1][1][-2]	[5][5][7]	17	848	63.63%
GA1-P-STS	0.393813	226	1	283.62	326.33	0.30	0.93	278.89	332.57	[2][0][0]	[5][5][6]	16	411	63.24%
GA4-P2	0.391778	170	1	318.54	334.75	0.41	0.68	310.75	337.49	[3][1][1]	[7][6][7]	20	3556	62.91%
GA3-G	0.388644	123	1	315.92	331.13	0.32	0.57	308.38	337.43	[2][0][1]	[5][8][7]	20	1488	62.41%
GA2-G-AO-H2	0.382648	804	10	318.32	338.40	0.15	0.27	284.78	347.02	[1][-1][0]	[7][8][7]	22	214	61.44%
GA4-P-H2	0.382379	475	6	277.08	336.31	0.21	0.77	276.64	336.62	[2][-1][-1]	[8][7][7]	22	1254	61.40%
GA1-G-H-STS	0.380821	509	7	296.03	310.43	0.13	0.29	292.35	317.85	[0][-1][1]	[8][6][8]	22	119	61.15%
GA2-G	0.380175	211	3	315.48	334.50	0.22	0.84	312.18	336.74	[3][-1][-1]	[5][8][6]	19	767	61.05%
GA3-P-H	0.37739	678	11	298.21	299.65	0.08	0.27	284.56	326.64	[2][0][2]	[5][6][5]	16	399	60.60%
GA2-G-AO-H	0.37039	94	2	304.58	326.51	0.21	0.29	286.06	332.78	[1][-1][0]	[7][8][7]	22	199	59.47%
GA2-G-H2	0.35835	500	15	302.76	314.24	0.27	0.31	300.33	324.90	[3][2][0]	[5][5][6]	16	199	57.54%
GA2-G2	0.357932	33	1	285.50	324.84	0.44	0.68	275.71	328.01	[1][2][-1]	[6][5][5]	16	843	57.47%
GA1-G	0.350493	28	1	296.47	321.22	0.49	0.79	279.39	312.30	[1][3][1]	[5][8][5]	18	398	56.28%
GA1-G-H	0.338444	337	15	296.28	314.86	0.07	0.30	277.45	324.96	[0][-2][2]	[8][6][7]	21	109	54.34%
GA4-P-AO-H	0.265184	1548	153	305.01	333.42	0.15	0.91	297.03	339.97	[1][-1][0]	[7][8][7]	22	756	42.58%
GA2-G-H	0.139572	41	8	286.99	297.47	0.35	0.40	278.14	338.18	[-2][2][1]	[6][6][5]	17	205	22.41%
GA1-P-AO	0	0	0	329.65	339.97	0.76	0.82	304.44	313.62	[1][-1][0]	[7][8][7]	22	377	0.00%

Table 14: Fuzzy-Genetic Colour Calibration Result for Purple

Experiment Name	Score	Hits	Misses	Angle		Radius		Contrast Angle		Contrast Rule	Colour Depth	bits	Running Time	Score
				Min	Max	Min	Max	Min	Max	[R][G][B]	[R][G][B]		(Sec)	Improvement
GA3-G	0.624469	3246	373	275.92	315.25	0.10	0.21	275.46	315.18	[0][0][-3]	[5][6][5]	16	1585	108.38%
GA1-P2	0.62104	2954	272	272.02	310.13	0.18	0.30	271.86	315.10	[1][1][-1]	[5][7][7]	19	407	107.79%
GA1-G-AO2	0.611166	3205	394	282.36	308.20	0.18	0.33	278.07	312.10	[0][1][-3]	[6][7][7]	20	468	106.07%
GA2-G-AO2	0.609407	3293	435	281.89	308.44	0.18	0.33	276.48	312.42	[0][1][-3]	[6][7][7]	20	868	105.77%
GA3-G-STS	0.608482	2916	285	284.95	309.97	0.24	0.39	284.80	311.89	[1][2][-1]	[6][8][6]	20	1694	105.61%
GA2-P-STS2	0.606669	3226	415	275.23	311.58	0.10	0.25	271.94	312.04	[0][0][-2]	[5][5][7]	17	851	105.29%
GA4-P-STS	0.606551	3084	353	282.12	310.36	0.28	0.39	280.89	320.00	[1][2][-2]	[8][8][5]	21	3480	105.27%
GA1-P-STS	0.605376	3061	348	282.27	308.98	0.30	0.71	277.99	311.35	[1][2][-1]	[7][5][6]	18	425	105.07%
GA3-P-AO	0.596829	2997	344	287.05	306.93	0.17	0.35	281.89	312.10	[0][1][-3]	[6][7][7]	20	1614	103.58%
GA3-P2	0.596469	2882	306	271.94	309.74	0.19	0.56	271.33	309.97	[1][1][-1]	[6][5][8]	19	1613	103.52%
GA2-G	0.596084	2944	330	275.54	310.05	0.10	0.26	275.61	310.13	[0][0][-2]	[5][5][8]	18	818	103.45%
GA1-P-AO-STS	0.591935	2892	318	281.41	308.28	0.18	0.59	276.16	308.60	[0][1][-3]	[6][7][7]	20	436	102.73%
GA2-P-AO-H-STS	0.589489	2859	307	284.91	306.69	0.17	0.32	282.05	311.07	[0][1][-3]	[6][7][7]	20	228	102.31%
GA4-P-AO-STS	0.589411	3028	376	283.80	306.61	0.18	0.54	281.57	311.78	[0][1][-3]	[6][7][7]	20	3624	102.30%
GA3-P-AO-STS	0.588913	2880	320	283.40	307.25	0.17	0.79	281.89	309.08	[0][1][-3]	[6][7][7]	20	1672	102.21%
GA2-G-AO-H2	0.58777	2725	266	283.80	307.89	0.16	0.34	276.72	307.97	[0][1][-3]	[6][7][7]	20	223	102.01%
GA2-G-AO-STS	0.587678	3270	482	281.81	307.65	0.18	0.35	264.16	312.42	[0][1][-3]	[6][7][7]	20	941	102.00%
GA3-P-STS	0.587112	2661	244	283.80	309.13	0.27	0.93	276.99	309.13	[1][2][-1]	[8][7][7]	22	1684	101.90%
GA1-P-AO-H	0.583651	2666	255	283.80	307.49	0.17	0.90	282.52	307.97	[0][1][-3]	[6][7][7]	20	108	101.30%
GA2-P-AO-STS	0.583548	2759	289	281.17	307.25	0.19	0.77	262.65	308.60	[0][1][-3]	[6][7][7]	20	852	101.28%
GA1-G-AO-H-STS2	0.583227	2598	234	283.80	307.65	0.18	0.76	273.70	307.97	[0][1][-3]	[6][7][7]	20	153	101.22%
GA1-P-H-STS	0.582836	2845	316	281.58	309.21	0.27	0.65	281.43	309.90	[1][2][-2]	[8][5][6]	19	108	101.16%
GA3-P-AO-H	0.582108	2676	260	287.05	307.41	0.14	0.34	287.21	308.52	[0][1][-3]	[6][7][7]	20	420	101.03%
GA3-G-AO-STS	0.581292	2527	215	286.58	307.89	0.20	0.71	270.76	308.60	[0][1][-3]	[6][7][7]	20	1842	100.89%
GA4-P-AO-H-STS	0.579975	2767	293	287.05	308.44	0.14	0.32	282.92	309.16	[0][1][-3]	[6][7][7]	20	904	100.66%
GA3-P-H-STS	0.579667	2975	373	280.13	306.45	0.32	0.93	279.21	311.04	[1][3][-2]	[5][7][8]	20	433	100.61%
GA4-P-AO-H2	0.578961	2663	262	287.05	307.25	0.14	0.34	285.70	308.52	[0][1][-3]	[6][7][7]	20	864	100.48%
GA3-G-AO-H	0.578611	2661	262	287.05	307.25	0.14	0.34	286.02	308.52	[0][1][-3]	[6][7][7]	20	427	100.42%
GA2-P-AO	0.578581	2823	318	288.57	306.61	0.15	0.42	276.48	311.07	[0][1][-3]	[6][7][7]	20	772	100.42%
GA1-P-AO-H-STS	0.578347	2547	222	287.29	307.57	0.15	0.31	285.46	308.60	[0][1][-3]	[6][7][7]	20	112	100.38%
GA2-P-H-STS	0.577585	3182	467	285.64	306.84	0.16	0.64	272.48	312.34	[0][1][-2]	[7][6][6]	19	211	100.24%
vLUT op	0.576178	2782	309	286.52	307.86	0.13	1.00	285.01	308.56	[0][1][-3]	[6][7][7]	20		100.00%
Reference	0.572888	2777	314	286.52	307.86	0.13	1.00	285.01	308.56	[0][1][-3]	[8][8][8]	24		99.43%
GA3-P-AO-H-STS	0.567061	2857	354	287.05	306.77	0.12	0.32	285.70	310.99	[0][1][-3]	[6][7][7]	20	443	98.42%
GA2-G-AO-H-STS	0.566613	3212	505	286.34	306.93	0.15	0.82	273.38	314.09	[0][1][-3]	[6][7][7]	20	244	98.34%
GA1-G	0.563693	2608	275	290.61	305.23	0.25	0.43	275.84	311.27	[1][2][0]	[6][7][5]	18	399	97.83%
GA1-G-AO-STS	0.561406	3110	471	287.77	306.85	0.14	0.36	279.74	313.93	[0][1][-3]	[6][7][7]	20	471	97.44%
GA3-G-H-STS2	0.557147	2984	425	285.49	310.89	0.21	0.39	273.70	311.20	[1][2][-1]	[7][8][6]	21	586	96.70%
GA3-G-AO-H-STS	0.55364	2420	233	285.15	306.85	0.15	0.45	284.99	307.25	[0][1][-3]	[6][7][7]	20	490	96.09%
GA2-P-H2	0.544741	3042	477	276.00	318.31	0.08	0.25	271.63	320.30	[1][0][-3]	[8][8][5]	21	278	94.54%
GA1-G-STS	0.536819	2863	429	287.40	305.84	0.31	0.96	289.69	313.80	[0][2][-2]	[5][6][5]	16	424	93.17%
GA1-G-AO-H2	0.534162	3501	732	284.83	307.41	0.14	0.91	282.44	317.66	[0][1][-3]	[6][7][7]	20	120	92.71%
GA2-P-H	0.488258	2845	527	287.63	308.98	0.19	0.81	288.24	310.58	[-3][1][0]	[6][6][7]	19	206	84.74%
GA4-P-AO2	0.480991	2783	516	287.21	304.78	0.13	0.57	274.89	316.47	[0][1][-3]	[6][7][7]	20	3336	83.48%
GA4-P-H-STS2	0.462845	1884	223	289.08	308.52	0.10	0.35	285.03	315.48	[1][0][0]	[6][5][5]	16	1159	80.33%
GA3-G-H2	0.450832	2221	350	280.13	309.67	0.09	0.21	280.28	323.36	[1][0][0]	[8][6][8]	22	455	78.25%
GA1-G-AO	0.428186	1888	274	289.44	303.04	0.13	0.44	283.40	311.70	[0][1][-3]	[6][7][7]	20	470	74.31%
GA3-G-AO	0.411815	3147	861	284.67	313.29	0.19	0.80	257.08	312.58	[0][1][-3]	[6][7][7]	20	1677	71.47%
GA2-G-AO	0.389417	132	1	299.86	305.90	0.21	0.23	258.51	309.00	[0][1][-3]	[6][7][7]	20	872	67.59%
GA2-G-STS2	0.386828	212	2	281.66	288.32	0.27	0.72	277.30	300.87	[-2][2][1]	[5][7][8]	20	854	67.14%
GA3-G-AO2	0.386828	106	1	286.98	294.93	0.29	0.31	278.71	306.93	[0][1][-3]	[6][7][7]	20	1724	67.14%
GA3-P	0.385307	95	1	317.47	318.62	0.23	0.26	281.97	319.77	[3][0][-2]	[7][7][8]	22	1921	66.87%
GA4-P	0.383774	86	1	315.64	318.70	0.32	0.43	307.37	336.53	[2][2][3]	[5][6][5]	16	3164	66.61%
GA2-P	0.381153	74	1	314.26	320.00	0.07	0.12	285.33	348.23	[2][-1][-3]	[5][7][6]	18	894	66.15%
GA1-P-AO	0.380632	72	1	289.36	291.35	0.29	0.88	285.07	308.04	[0][1][-3]	[6][7][7]	20	413	66.06%
GA4-P2	0.379793	69	1	328.49	336.91	0.10	0.15	273.70	343.95	[0][0][3]	[8][8][5]	21	3547	65.92%
GA4-P-AO-H	0.375831	2330	531	285.07	303.19	0.18	0.58	285.31	326.73	[0][1][-3]	[6][7][7]	20	1002	65.23%
GA2-P-STS	0.373725	53	1	314.18	328.49	0.61	0.74	299.95	336.99	[2][2][3]	[6][7][5]	18	877	64.86%
GA2-P2	0.37039	94	2	287.86	312.42	0.45	0.86	278.75	301.86	[3][3][-2]	[6][6][5]	17	886	64.28%
GA2-G-H	0.362039	549	15	282.58	295.05	0.33	0.47	277.37	301.71	[-1][3][0]	[8][7][5]	20	213	62.83%
GA4-P-AO	0.358133	3591	1294	273.78	312.89	0.18	0.37	260.10	312.02	[0][1][-3]	[6][7][7]	20	3758	62.16%
GA2-G-STS	0.352184	58	2	296.28	330.71	0.66	0.87	303.85	344.79	[-1][2][3]	[7][7][6]	20	838	61.12%
GA2-P-AO-H	0.347579	370	14	290.55	296.75	0.25	0.83	269.72	307.33	[0][1][-3]	[6][7][7]	20	214	60.32%
GA1-G-H-STS2	0.346567	337	13	287.32	299.11	0.13	0.37	283.50	301.10	[1][0][2]	[5][5][7]	17	150	60.15%
GA1-G-H-STS	0.336441	87	4	286.56	294.52	0.14	0.55	271.56	342.27	[2][0][2]	[6][8][8]	22	125	58.39%
GA4-P-H-STS	0.327098	265	14	293.37	297.50	0.06	0.30	292.22	301.71	[0][0][1]	[7][6][7]	20	853	56.77%
GA1-G-H	0.317099	731	44	294.59	301.56	0.38	0.58	278.83	307.52	[1][3][0]	[8][5][8]	21	117	55.03%
GA2-G-H-STS	0.309699	396	26	290.77	305.23	0.19	0.85	286.94	305.08	[2][1][0]	[6][5][6]	17	239	53.75%
GA2-P-AO-H2	0.296632	146	11	290.16	292.78	0.30	0.74	268.61	312.10	[0][1][-3]	[6][7][7]	20	235	51.48%
GA4-P-H	0.294779	2946	1028	274.24	309.67	0.21	0.54	272.09	321.22	[2][2][0]	[8][6][5]	19	1175	51.16%
GA3-G-H	0.278458	225	20	291.53	301.56	0.10	0.80	288.16	313.19	[2][0][1]	[5][6][6]	17	399	48.33%
GA2-G-H-STS2	0.275488	1800	164	283.11	305.23	0.19	0.93	281.58	305.08	[0][1][-2]	[6][5][7]	18	304	47.81%
GA3-P-H2	0.2704	1359	129	274.70	308.60	0.34	0.43	292.60	313.88	[2][2][0]	[7][8][8]	23	659	46.93%
GA3-G-H-STS	0.270221	263	25	290.77	300.64	0.29	0.58	273.47	303.54	[3][2][2]	[6][5][6]	17	462	46.90%
GA4-P-H2	0.267272	586	57	278.29	300.33	0.34	0.79	278.06	307.68	[2][3][0]	[7][7][7]	21	958	46.39%
GA1-G-AO-H	0.248067	358	40	301.45	302.80	0.14	0.85	279.18	336.91	[0][1][-3]	[6][7][7]	20	110	43.05%
GA1-G-H2	0.235829	281	34	309.44	317.47	0.09	0.18	271.94	319.92	[2][0][-2]	[7][6][5]	18	118	40.93%
GA1-G-AO-H-STS	0.223227	1861	243	285.39	303.19	0.13	1.00	276.80	307.97	[0][1][-3]	[6][7][7]	20	125	38.74%
GA2-G-H2	0.18104	178	29	317.93	325.97	0.31	0.45	290.84	346.70	[0][2][3]	[8][8][5]	21	236	31.42%
GA3-P-H	0.167291	121	21	334.08	336.60	0.14	0.29	293.67	344.25	[0][0][2]	[8][8][7]	23	420	29.03%
GA2-G-AO-H	0.133333	135	27	295.88	297.47	0.21	0.89	281.57	320.37	[0][1][-3]	[6][7][7]	20	228	23.14%
GA1-P	0	0	0	285.79	295.74	0.62	0.79	273.62	313.95	[0][3][0]	[7][6][5]	18	432	0.00%
GA1-P-AO2	0	0	0	283.40	295.80	0.58	0.62	258.43	336.83	[0][1][-3]	[6][7][7]	20	460	0.00%
GA1-P-H	0	0	0	274.47	287.17	0.48	0.60	274.70	296.58	[1][-2][-2]	[6][6][7]	19	117	0.00%
GA1-P-H2	0	0	0	333.85	336.30	0.15	0.15	271.48	288.32	[2][-2][0]	[7][7][7]	21	144	0.00%

Table 15: Fuzzy-Genetic Colour Calibration Result for Violet

Experiment Name	Score	Hits	Misses	Angle		Radius		Contrast Angle		Contrast Rule	Colour Depth	bits	Running Time	Score
				Min	Max	Min	Max	Min	Max	[R][G][B]	[R][G][B]		(Sec)	Improvement
GA2-G-H2	0.72671	2312	43	215.80	262.42	0.07	0.51	215.52	289.69	[0][0][2]	[7][8][8]	23	261	120.52%
GA3-P2	0.718614	2401	85	200.79	275.70	0.05	0.12	193.00	292.72	[0][0][1]	[6][7][7]	20	1931	119.18%
GA4-P	0.716794	2370	79	197.19	274.97	0.05	0.12	195.89	290.42	[0][0][1]	[8][5][6]	19	3896	118.88%
GA2-G2	0.716699	2386	83	197.04	275.70	0.05	0.12	190.40	293.73	[0][0][1]	[6][7][6]	19	922	118.86%
GA3-G	0.713432	2392	92	199.35	275.70	0.05	0.13	189.97	291.14	[0][0][1]	[6][7][7]	20	1708	118.32%
GA2-G-H-STS2	0.71	2363	88	203.39	274.25	0.05	0.11	200.36	291.57	[0][0][1]	[6][7][8]	21	316	117.75%
GA3-G-H-STS	0.703227	2435	120	217.97	267.61	0.07	0.71	206.13	326.49	[0][0][2]	[8][6][7]	21	510	116.63%
GA4-P-H-STS	0.698171	2330	100	209.89	268.48	0.06	0.74	210.03	293.73	[0][1][2]	[8][7][8]	23	969	115.79%
GA2-P-STS	0.686543	2264	105	198.20	271.08	0.06	0.48	195.17	294.17	[0][1][2]	[7][8][6]	21	969	113.86%
GA3-P-STS2	0.684712	2467	166	217.39	272.95	0.04	0.26	198.49	312.35	[0][0][1]	[5][7][7]	19	1855	113.55%
GA4-P-STS	0.681991	2166	79	217.82	267.47	0.06	0.85	211.04	291.14	[0][1][2]	[6][7][7]	20	3690	113.10%
GA3-G-STS	0.671357	2420	171	228.07	275.26	0.04	0.13	224.17	302.10	[0][0][1]	[7][5][6]	18	1793	111.34%
GA3-P-AO	0.668382	2128	90	205.99	276.86	0.04	0.23	205.55	290.62	[0][1][1]	[5][7][5]	17	1519	110.85%
GA1-P-STS	0.668079	2013	60	214.79	252.89	0.09	0.26	196.90	289.84	[-1][0][3]	[6][7][8]	21	476	110.80%
GA4-P-AO-H	0.666476	2154	100	207.74	277.15	0.04	0.29	205.69	297.07	[0][1][1]	[5][7][5]	17	820	110.53%
GA2-G-AO2	0.666169	2202	116	197.06	276.86	0.04	0.26	196.32	292.53	[0][1][1]	[5][7][5]	17	841	110.48%
GA2-G-STS	0.666113	1858	26	209.31	257.22	0.07	0.32	199.64	279.30	[0][0][2]	[7][6][6]	19	940	110.47%
GA2-P-AO2	0.660799	2182	119	219.90	279.93	0.04	0.23	220.34	297.07	[0][1][1]	[5][7][5]	17	802	109.59%
GA4-P-AO	0.66031	2017	73	219.90	277.15	0.04	0.23	220.34	290.62	[0][1][1]	[5][7][5]	17	2959	109.51%
GA3-P-AO-H-STS	0.659747	2107	96	208.18	275.69	0.04	0.20	205.69	299.56	[0][1][1]	[5][7][5]	17	410	109.41%
GA1-P-AO-STS	0.659698	2143	105	206.43	275.39	0.04	0.25	206.13	292.53	[0][1][1]	[5][7][5]	17	392	109.41%
GA3-G-AO	0.659502	2185	122	219.90	279.93	0.04	0.29	219.46	297.07	[0][1][1]	[5][7][5]	17	1580	109.37%
GA4-P-AO-H-STS	0.655809	2272	153	204.52	277.15	0.04	0.31	202.18	297.80	[0][1][1]	[5][7][5]	17	806	108.76%
GA2-P-AO-STS	0.652535	2059	92	203.94	273.93	0.04	0.30	203.35	304.68	[0][1][1]	[5][7][5]	17	783	108.22%
GA1-P2	0.651976	2037	88	217.68	254.48	0.06	0.47	218.26	284.50	[0][0][2]	[5][5][8]	18	471	108.13%
GA2-P-H-STS	0.650707	2240	149	219.99	234.85	0.05	0.50	216.96	289.55	[-3][-1][2]	[5][6][5]	16	238	107.92%
GA2-G-AO-H2	0.65038	1995	80	220.04	275.83	0.04	0.16	219.02	297.51	[0][1][1]	[5][7][5]	17	218	107.86%
GA1-P-AO-H-STS2	0.648989	2206	144	194.86	275.83	0.04	0.28	191.34	321.52	[0][1][1]	[5][7][5]	17	141	107.63%
GA3-P-AO-STS	0.646273	2022	95	220.92	277.15	0.04	0.23	217.70	291.94	[0][1][1]	[5][7][5]	17	1498	107.18%
GA2-G-AO-STS	0.644569	2125	128	202.62	277.30	0.04	0.17	196.76	309.37	[0][1][1]	[5][7][5]	17	814	106.90%
GA3-P-AO-H	0.643838	1919	70	222.24	275.98	0.04	0.46	222.39	291.06	[0][1][1]	[5][7][5]	17	385	106.78%
GA4-P-AO-STS	0.636728	2015	109	227.07	279.35	0.04	0.66	221.51	303.22	[0][1][1]	[5][7][5]	17	3161	105.60%
GA1-G-AO2	0.636617	2321	200	199.98	275.98	0.03	0.24	198.67	297.07	[0][1][1]	[5][7][5]	17	451	105.58%
GA1-G-AO-STS	0.632555	1770	49	219.90	272.61	0.04	0.29	215.21	326.50	[0][1][1]	[5][7][5]	17	425	104.90%
GA1-P-AO	0.628277	2098	145	224.73	278.76	0.03	0.23	220.48	292.23	[0][1][1]	[5][7][5]	17	357	104.20%
GA1-G2	0.619765	2144	169	229.66	254.48	0.07	0.74	204.84	289.55	[0][0][3]	[7][5][6]	18	477	102.78%
GA2-P-AO-H-STS	0.619303	2132	166	219.90	275.83	0.03	0.32	219.17	301.02	[0][1][1]	[5][7][5]	17	207	102.71%
GA2-G-AO-H-STS	0.609073	2116	180	226.63	279.35	0.03	0.62	222.53	300.87	[0][1][1]	[5][7][5]	17	221	101.01%
GA2-P-AO-H2	0.604678	2368	274	199.54	275.98	0.03	0.25	198.81	311.42	[0][1][1]	[5][7][5]	17	200	100.28%
vLUT op	0.602979	1802	101	232.34	282.28	0.04	1.00	228.31	293.36	[0][1][1]	[5][7][5]	17		100.00%
GA3-G-AO-H-STS	0.599224	1928	136	225.02	274.66	0.04	0.56	197.35	313.90	[0][1][1]	[5][7][5]	17	436	99.38%
GA1-G-AO-H-STS	0.599217	1905	135	227.81	281.25	0.04	0.72	213.75	315.52	[0][1][1]	[5][7][5]	17	112	99.38%
GA3-G-AO-STS	0.597606	1934	141	230.73	276.57	0.03	0.81	218.58	314.05	[0][1][1]	[5][7][5]	17	1525	99.11%
GA1-G-H2	0.591937	2065	183	217.10	291.28	0.04	0.08	217.54	293.45	[0][0][1]	[8][8][6]	22	117	98.17%
GA3-P-H2	0.587096	2233	253	228.65	254.34	0.08	0.74	214.65	305.28	[0][0][3]	[7][6][6]	19	551	97.37%
GA3-G-AO-H	0.585199	1943	161	229.27	275.83	0.03	0.75	220.78	310.68	[0][1][1]	[5][7][5]	17	439	97.05%
GA3-P-H-STS	0.566585	1776	138	229.95	247.55	0.06	0.80	204.26	281.61	[0][0][3]	[7][6][6]	19	469	93.96%
Reference	0.526654	2535	497	232.34	282.28	0.04	1.00	228.31	293.36	[1][1][2]	[8][8][8]	24		87.34%
GA1-G-STS	0.50063	2506	533	198.34	288.25	0.07	0.47	191.13	315.82	[1][0][2]	[8][8][6]	22	457	83.03%
GA2-G-H-STS	0.415997	2423	663	215.23	282.62	0.04	0.92	202.82	317.69	[1][2][2]	[7][5][6]	18	265	68.99%
GA2-G	0.398163	762	1	255.20	296.04	0.05	0.14	220.13	299.07	[1][-2][0]	[5][6][6]	17	917	66.03%
GA3-P-STS	0.397896	665	1	219.99	249.57	0.11	0.89	200.22	271.80	[-2][-1][3]	[7][7][6]	20	1830	65.99%
GA3-P	0.396761	432	1	255.78	268.33	0.15	0.69	222.15	290.56	[0][2][3]	[8][6][8]	22	1992	65.80%
GA2-G-H	0.393982	697	3	241.49	260.83	0.07	0.10	219.84	278.44	[0][-1][1]	[8][7][8]	23	266	65.34%
GA1-P-AO-H-STS	0.384349	535	6	257.38	272.32	0.06	0.28	227.51	328.11	[0][1][1]	[5][7][5]	17	103	63.74%
GA4-P-H2	0.383536	678	8	252.60	272.09	0.08	0.59	218.83	274.68	[1][0][3]	[8][5][8]	21	1012	63.61%
GA1-G-H	0.378389	129	2	230.52	252.75	0.13	0.49	204.98	280.17	[-2][3][2]	[5][5][8]	18	120	62.75%
GA1-P-H-STS2	0.377675	437	7	239.76	263.72	0.13	0.97	236.59	298.50	[0][2][3]	[7][7][8]	22	160	62.63%
GA2-G-AO-H	0.377208	428	7	245.96	266.90	0.07	0.60	209.21	291.21	[0][1][1]	[5][7][5]	17	196	62.56%
GA1-G-H-STS2	0.376224	293	5	208.59	239.76	0.02	0.19	190.84	303.84	[0][3][1]	[5][5][6]	16	163	62.39%
GA1-G	0.37556	57	1	227.64	268.62	0.04	0.17	225.04	303.98	[1][2][0]	[5][5][7]	17	399	62.28%
GA2-P-AO-H	0.371103	289	6	254.89	278.76	0.10	0.22	243.77	308.63	[0][1][1]	[5][7][5]	17	183	61.54%
GA2-G-AO	0.371089	337	7	210.38	240.69	0.05	0.45	197.35	312.73	[0][1][1]	[5][7][5]	17	764	61.54%
GA1-G-AO-H	0.370597	142	3	245.08	258.26	0.06	0.56	197.93	262.22	[0][1][1]	[5][7][5]	17	110	61.46%
GA2-P-AO	0.355242	31	1	265.00	280.37	0.15	0.61	242.30	318.88	[0][1][1]	[5][7][5]	17	710	58.91%
GA1-G-H-STS	0.342333	480	20	273.96	287.53	0.14	0.69	251.59	306.87	[0][3][1]	[5][6][5]	16	129	56.77%
GA1-G-AO	0.34178	309	13	267.63	281.40	0.09	0.28	260.46	295.75	[0][1][1]	[5][7][5]	17	369	56.68%
GA3-G-H2	0.338534	180	8	222.44	284.93	0.18	0.25	196.47	321.88	[0][3][1]	[5][6][7]	18	454	56.14%
GA2-P	0.287499	1047	86	266.03	297.05	0.10	0.35	194.88	305.42	[3][0][2]	[6][8][8]	22	984	47.68%
GA3-G-H	0.287428	292	24	204.11	275.98	0.03	0.14	182.32	295.76	[-1][3][0]	[8][7][8]	23	448	47.67%
GA3-P-H	0.2646	967	96	205.41	287.24	0.10	0.14	191.99	295.47	[1][1][3]	[5][5][8]	18	459	43.88%
GA4-P-H	0.213584	319	44	251.02	254.91	0.03	0.75	207.72	310.19	[0][0][1]	[8][5][5]	18	1111	35.42%
GA1-P-H-STS	0.172537	59	10	212.05	250.30	0.21	0.33	204.40	309.75	[2][3][3]	[8][5][5]	18	121	28.61%
GA1-G-AO-H2	0	0	0	315.37	322.69	0.56	0.59	197.64	252.26	[0][1][1]	[5][7][5]	17	101	0.00%
GA1-P	0	0	0	261.26	293.30	0.59	0.97	218.26	268.33	[3][2][2]	[7][7][8]	22	462	0.00%
GA1-P-AO-H	0	0	0	252.26	259.87	0.46	0.76	274.52	329.43	[0][1][1]	[5][7][5]	17	92	0.00%
GA1-P-AO-H2	0	0	0	237.32	282.72	0.40	0.45	195.15	301.90	[0][1][1]	[5][7][5]	17	96	0.00%
GA2-P2	0	0	2144	247.26	313.36	0.17	0.36	208.16	325.05	[2][0][1]	[7][5][5]	17	1040	0.00%
GA1-P-H	0	0	0	278.87	318.99	0.58	0.78	230.81	284.79	[0][-2][2]	[8][5][6]	19	115	0.00%
GA2-P-H	0	0	37	240.48	254.91	0.25	0.44	212.05	288.25	[0][1][1]	[7][6][5]	18	228	0.00%
GA2-P-H2	0	0	1347	258.52	295.90	0.34	0.58	267.76	288.39	[2][1][2]	[8][7][6]	21	313	0.00%
GA1-P-H2	-5640	1	5640	304.41	327.07	0.33	0.98	196.18	271.37	[2][0][0]	[5][5][7]	17	158	-935355.96%

Table 16: Fuzzy-Genetic Colour Calibration Result for Light Blue

Experiment Name	Score	Hits	Misses	Angle		Radius		Contrast Angle		Contrast Rule	Colour Depth	bits	Running Time	Score
				Min	Max	Min	Max	Min	Max	[R][G][B]	[R][G][B]		(Sec)	Improvement
GA3-P	0.724412	3016	17	142.97	189.69	0.07	0.17	130.30	189.92	[-1][0][1]	[8][6][6]	20	1778	104.85%
GA3-G-STS	0.719549	2988	21	144.19	182.82	0.07	0.26	129.99	187.55	[-1][0][1]	[7][8][5]	20	1724	104.15%
GA1-P-STS	0.712497	2907	15	139.53	182.97	0.08	0.19	136.10	184.42	[-1][0][1]	[5][6][6]	17	446	103.13%
GA3-P-STS	0.711634	2901	16	143.05	184.88	0.07	0.54	129.69	186.10	[-1][0][3]	[5][5][8]	18	1750	103.00%
GA4-P	0.710999	3010	44	136.33	156.10	0.05	0.35	129.15	184.19	[0][3][1]	[5][8][6]	19	3751	102.91%
GA2-G-AO	0.708627	2958	36	137.36	169.93	0.00	0.32	136.94	168.59	[0][3][1]	[5][5][6]	16	783	102.57%
GA4-P-AO-H	0.708099	2958	38	137.69	158.96	0.00	0.33	135.26	169.01	[0][3][1]	[5][5][6]	16	785	102.49%
GA4-P-AO	0.707892	2998	48	136.43	170.43	0.04	0.36	128.90	169.01	[0][3][1]	[5][5][6]	16	2948	102.46%
GA2-G-AO-H	0.707737	3005	50	136.52	170.85	0.04	0.96	128.56	169.01	[0][3][1]	[5][5][6]	16	205	102.44%
GA3-P-AO	0.707734	2993	47	136.43	170.43	0.04	0.82	128.56	168.51	[0][3][1]	[5][5][6]	16	1456	102.44%
GA3-G-AO2	0.706957	2995	50	137.77	160.30	0.00	0.42	135.09	172.69	[0][3][1]	[5][5][6]	16	1523	102.33%
GA3-P-H	0.706927	3057	67	120.22	152.44	0.06	0.66	114.04	186.03	[-1][3][-1]	[8][6][8]	22	461	102.32%
GA3-P-AO-H-STS	0.706774	2922	31	137.77	170.35	0.04	0.50	128.90	168.76	[0][3][1]	[5][5][6]	16	399	102.30%
GA3-G-H-STS	0.70624	2818	10	144.27	183.20	0.08	0.88	127.24	186.87	[-1][0][1]	[6][8][8]	22	461	102.22%
GA2-P	0.706064	3013	54	136.33	161.22	0.04	0.57	136.33	180.61	[0][3][1]	[5][8][6]	19	846	102.20%
GA1-P-AO-H	0.705609	3046	66	136.52	156.36	0.00	0.69	134.84	170.27	[0][3][1]	[5][5][6]	16	108	102.13%
GA1-P-AO-H-STS	0.704976	2950	43	137.94	161.39	0.00	0.51	134.84	168.93	[0][3][1]	[5][5][6]	16	101	102.04%
GA2-P-AO	0.704424	2921	36	137.77	160.38	0.04	0.51	127.22	169.60	[0][3][1]	[5][5][6]	16	755	101.96%
GA4-P-AO-STS	0.704278	3071	75	135.26	158.54	0.00	0.80	136.10	170.27	[0][3][1]	[5][5][6]	16	3057	101.94%
GA2-P-AO-H-STS	0.70377	2915	37	138.03	158.04	0.00	0.78	135.09	168.34	[0][3][1]	[5][5][6]	16	200	101.86%
GA1-P-AO-STS	0.703716	2932	40	137.52	169.26	0.04	0.48	127.56	169.93	[0][3][1]	[5][5][6]	16	378	101.86%
GA4-P-H	0.702536	2842	26	144.19	182.06	0.10	0.52	127.93	187.78	[-1][1][2]	[8][5][7]	20	876	101.69%
GA2-P-AO-STS	0.701825	2916	40	137.69	156.53	0.04	0.91	128.39	169.01	[0][3][1]	[5][5][6]	16	761	101.58%
GA3-P-AO-STS	0.700936	3089	88	135.35	171.02	0.01	0.63	135.09	172.95	[0][3][1]	[5][5][6]	16	1452	101.45%
GA3-G-H	0.700433	2872	34	147.47	170.45	0.09	0.39	130.60	174.50	[-1][1][3]	[6][6][7]	19	463	101.38%
GA1-G	0.6993	2871	33	132.43	176.79	0.07	0.28	131.37	190.23	[-1][3][0]	[5][8][5]	18	455	101.22%
GA3-G-AO-STS	0.697827	3047	82	133.75	162.39	0.04	0.56	129.90	168.76	[0][3][1]	[5][5][6]	16	1460	101.00%
GA2-P-H-STS	0.697702	2919	50	148.70	169.61	0.08	0.19	130.14	187.25	[-1][1][3]	[5][6][8]	19	223	100.99%
GA4-P-AO-H-STS	0.697685	2832	28	139.37	162.31	0.01	0.91	137.27	168.84	[0][3][1]	[5][5][6]	16	774	100.98%
GA3-G	0.696923	2985	69	149.08	172.90	0.04	0.51	129.23	174.50	[0][1][3]	[8][6][8]	22	1740	100.87%
GA1-G-AO-H-STS	0.696652	3057	88	137.77	163.90	0.02	0.84	130.99	182.32	[0][3][1]	[5][5][6]	16	109	100.83%
GA4-P-STS	0.6954	2887	47	141.37	171.29	0.08	0.44	131.60	180.45	[-1][1][3]	[7][5][7]	19	3452	100.65%
GA2-P-AO-H	0.695091	2820	33	137.69	159.88	0.06	0.90	130.32	179.56	[0][3][1]	[5][5][6]	16	191	100.61%
GA1-G-AO-STS	0.694844	3073	96	137.77	186.34	0.03	0.79	131.41	188.94	[0][3][1]	[5][5][6]	16	410	100.57%
GA2-G-AO-STS	0.694764	3167	125	135.43	161.98	0.01	0.55	134.42	181.82	[0][3][1]	[5][5][6]	16	796	100.56%
GA2-G-STS	0.694159	2943	63	153.05	169.84	0.08	0.33	132.43	188.24	[-1][1][3]	[6][8][8]	22	856	100.47%
GA4-P-H-STS	0.693609	2870	47	130.22	185.11	0.09	0.56	129.30	188.62	[-1][3][0]	[7][7][7]	21	942	100.39%
GA3-P-AO-H	0.693435	2805	33	138.03	165.58	0.06	0.79	127.81	178.81	[0][3][1]	[5][5][6]	16	379	100.37%
GA2-P-STS	0.692361	2837	44	138.85	173.35	0.06	0.53	136.33	185.19	[0][1][2]	[8][7][8]	23	908	100.21%
GA1-P2	0.691348	2956	71	148.01	158.77	0.12	0.93	134.42	187.63	[-2][2][3]	[5][5][8]	18	469	100.07%
vLUT op	0.690887	2786	30	137.12	162.79	0.02	1.00	136.94	163.51	[0][3][1]	[5][5][6]	16		100.00%
GA3-G-AO-H-STS	0.690554	2838	47	137.19	171.02	0.06	0.35	123.45	176.71	[0][3][1]	[5][5][6]	16	423	99.95%
GA2-G-AO-H-STS	0.690477	2842	48	137.19	161.47	0.06	0.65	126.05	179.39	[0][3][1]	[5][5][6]	16	221	99.94%
GA2-G-H	0.688664	2762	30	133.73	156.25	0.12	0.17	130.98	184.58	[-2][1][0]	[5][5][8]	18	252	99.68%
GA2-G-H-STS	0.687317	2887	64	151.98	170.68	0.02	0.35	130.22	171.83	[0][1][3]	[5][6][7]	18	242	99.48%
GA2-P-H	0.687245	2813	44	144.27	156.25	0.09	0.93	136.18	175.34	[-2][2][3]	[8][6][7]	21	226	99.47%
GA3-G-AO-H	0.686541	2722	25	139.45	167.08	0.05	0.99	130.49	169.76	[0][3][1]	[5][5][6]	16	383	99.37%
GA1-P-H	0.686472	2828	52	127.47	175.11	0.06	0.42	115.18	184.35	[0][3][0]	[8][7][8]	23	112	99.36%
GA1-P-AO	0.685425	2729	30	139.78	182.66	0.06	0.45	131.07	189.36	[0][3][1]	[5][5][6]	16	343	99.21%
GA1-P-H-STS	0.684	3013	103	135.56	159.23	0.12	0.93	131.60	173.58	[-3][2][3]	[5][5][8]	18	119	99.00%
GA2-G	0.681764	2668	28	138.85	189.31	0.14	0.25	130.07	189.62	[-2][2][3]	[7][8][8]	23	1024	98.68%
GA3-P-H-STS	0.676858	2869	81	148.39	159.31	0.02	0.66	138.16	189.08	[1][2][3]	[7][8][6]	21	441	97.97%
GA1-G-STS	0.672695	2776	64	142.28	175.87	0.00	0.17	132.51	182.06	[0][3][1]	[7][5][8]	20	429	97.37%
Reference	0.668808	2758	68	137.12	162.79	0.02	1.00	136.94	163.51	[0][3][1]	[8][8][8]	24		96.80%
GA1-G-AO-H	0.648037	2494	50	141.29	153.85	0.00	0.56	137.36	177.13	[0][3][1]	[5][5][6]	16	100	93.80%
GA1-G-AO	0.647494	2579	63	143.72	180.23	0.01	0.73	119.60	180.23	[0][3][1]	[5][5][6]	16	383	93.72%
GA1-G-H-STS2	0.595834	3201	369	142.13	164.88	0.08	0.84	129.69	179.46	[-1][2][3]	[6][6][8]	20	157	86.24%
GA1-G-H	0.506662	2641	353	145.64	155.03	0.07	0.90	137.86	165.80	[-1][1][1]	[5][7][5]	17	123	73.34%
GA1-P	0.399055	1481	1	127.85	157.63	0.16	0.98	123.12	168.39	[-2][3][-1]	[5][8][8]	21	439	57.76%
GA3-G-AO	0.393113	203	1	134.01	141.96	0.13	0.43	131.07	167.25	[0][3][1]	[5][5][6]	16	1433	56.90%
GA1-G-H-STS	0.390267	2584	18	146.56	176.64	0.11	0.41	137.32	178.93	[-2][0][2]	[5][7][6]	18	130	56.49%

Table 17: Colour Contrast Rule Component Distribution

Yellow Levels		Contrast Rules Red	Green	Blue		Green Levels		Contrast Rules Red	Green	Blue
	-3	0	0	0			-3	0	0	0
Degrade	-2	11	2	7		Degrade	-2	0	2	9
	-1	0	8	7			-1	0	4	2
Neutral	0	3	2	4		Neutral	0	6	3	0
	1	1	2	1			1	3	2	0
Enhance	2	0	1	0		Enhance	2	2	0	0
	3	4	4	0			3	0	0	0

Pink Levels		Contrast Rules Red	Green	Blue		Purple Levels		Contrast Rules Red	Green	Blue
	-3	0	0	2			-3	0	0	1
Degrade	-2	0	3	3		Degrade	-2	0	0	6
	-1	0	5	10			-1	0	0	5
Neutral	0	3	0	4		Neutral	0	4	3	0
	1	12	1	0			1	8	3	0
Enhance	2	4	0	0		Enhance	2	0	5	0
	3	0	10	0			3	0	1	0

Violet Levels		Contrast Rules Red	Green	Blue		Light Blue Levels		Contrast Rules Red	Green	Blue
	-3	1	0	0			-3	0	0	0
Degrade	-2	0	0	0		Degrade	-2	1	0	0
	-1	1	1	0			-1	13	0	1
Neutral	0	15	13	0		Neutral	0	4	5	2
	1	0	3	7			1	0	7	6
Enhance	2	0	0	8		Enhance	2	0	1	2
	3	0	0	2			3	0	5	7

Table 18: Colour Pixel Distribution Changes after FCCF Applied in 6 Target Colours

	Yellow	Green	Pink	Purple	Violet	Light Blue
Total Target Pixels	4,649	6,637	4,221	5,432	4,102	5,678
Number of Pixels within Colour Contrast Angles	2,372	4,191	3,972	3,772	2,976	3,620
Number of Pixels within Colour Contrast Angles after FCCF Processed	2,266	3,249	3,144	2,917	2,234	2,829
Number of Pixels within Colour Classification Angles	56	4,225	804	3,538	2,516	3,557
Number of Pixels within Colour Classification Angles after FCCF Processed	2,261	3,249	2,402	2,822	1,999	2,821
Number of Pixels moved in Colour Classification Angle after FCCF Processed	2,205	- 976	1,598	- 716	- 517	- 736

Table 19: Colour Classification and Contrast Angle Difference Between Fuzzy-Genetic Optimised Solution and Manual Calibrated Solution (size difference represents the angle difference relative to the base)

		Yellow	**Green**	**Pink**	**Purple**	**Violet**	**Light Blue**
	Min	43.992	45.576	314.424	286.524	232.344	137.124
Base	Max	46.476	96.66	327.276	307.859	282.276	162.792
	Angle Size	**2.484001**	**51.084**	**12.85199**	**21.33502**	**49.93201**	**25.66801**
Classification Angle	Min	32.1847	44.5325	283.437	275.92	215.803	142.97
Best	Max	52.4194	101.227	335.434	315.252	262.418	189.691
	Size Difference	**17.7507**	**5.610496**	**39.14501**	**17.99698**	**-3.31701**	**21.05299**
	Min	41.832	45.288	275.256	285.012	228.312	136.944
Base	Max	47.808	96.064	331.884	308.556	293.364	163.512
	Angle Size	**5.975998**	**50.77601**	**56.62799**	**23.54401**	**65.05202**	**26.56799**
Contrast Angle	Min	15.7038	37.9468	280.507	275.46	215.515	130.297
Best	Max	67.1408	121.461	335.371	315.176	289.693	189.92
	Size Difference	**45.461**	**32.7382**	**-1.76399**	**16.17199**	**9.125982**	**33.05501**

Chapter 6

Conclusions

The course of experimentation and exploration was initiated out of simple curiosity; that colour classification accuracy would be affected when the colour resolution is changed. This has led to the development of the Variable Colour Depth algorithm which proves to be an outstanding addition to the myriad of algorithms we employ for colour classification. This not only reduces memory consumption but also improves colour classification accuracy. In this case, we modified the test images into their lesser colour depth representation, such as 21-bits, 18-bits and 15-bits to test whether FCCF could compensate for the loss of information and maintain high colour classification accuracy. Interestingly, the results obtained in this research are remarkable. The battery of experimentations that we have gone through was quite an adventure. As a consequence of this research, generating colour classification parameters is now an automatic process which yields higher accuracy and lesser memory consumption.

We have presented a new approach to improving colour discriminability down to the bit level. We have introduced the Variable Colour Depth algorithm, along with accompanying techniques for building and searching a VCD LUT. We have fused to VCD algorithm with FCCF and extended CCRE and tested it against to the FCCF and CCRE combination to prove its efficiency. The results of experiments show that there is an increase of 6.9% in terms of overall colour classification accuracy and reduced memory space by 91.99%. Lastly, we incorporated the HAGA algorithm for fully automatically calibrating the parameters and improving overall colour classification accuracy by further 9.82%.

6.1 Suggestions for Future Work

There are three areas that can be identified for further work;

1. Analysis of colour classification in an outdoor environment and colour constancy emulation to compensate for changing ambient illumination. We believe that the results of the proposed study could introduce a universal colour classification technique both for indoor and outdoor use.

2. Extension of colour classification to make it work in the infrared (IR) and ultra violet (UV) spectrums.

3. Development of extended colour contrast rules customised for colour correction to aid the colour blind. Varying levels of colour blindness could be taken into account.

Bibliography

[1] Reyes, N.H., Messom, C.: Identifying colour objects with fuzzy colour contrast fusion. In: 3rd International Conference on Computational Intelligence, Robotics and Autonomous Systems, and FIRA RoboWorld Congress. (2005)

[2] Adelson, E.: Lightness perception and lightness illusions (1999)

[3] Smith, T., Guild, J.: The c.i.e. colorimetric standards and their use. Transactions of the Optical Society **33**(3) (1931) 73–134

[4] Dong, L., Ogunbona, P., Li, W., Yu, G., Fan, L., Zheng, G.: A fast algorithm for color image segmentation. In: Proceedings of the First International Conference on Innovative Computing, Information and Control-Volume 2. (2006) 685–688

[5] Guerrero, P., Ruiz-del Solar, J., Fredes, J., Palma-Amestoy, R.: Automatic online color calibration using class-relative color spaces. In Visser, U And Ribeiro, F And Ohashi, T And Dellaert, F, ed.: Robocup 2007: Robot Soccer World Cup Xi. Volume 5001 of Lecture Notes In Computer Science., Heidelberger Platz 3, D-14197 Berlin, Germany, Springer-Verlag Berlin (2008) 246–253 11th RoboCup International Symposium, Atlanta, GA, JUL 09-10, 2007.

[6] Takahashi, Y., Nowak, W., Wisspeintner, T.: Adaptive recognition of color-coded objects in indoor and outdoor environments. In Visser, U And Ribeiro, F And Ohashi, T And Dellaert, F, ed.: Robocup 2007: Robot Soccer World Cup Xi. Volume 5001 of Lecture Notes In Computer Science., Heidelberger

Platz 3, D-14197 Berlin, Germany, Springer-Verlag Berlin (2008) 65–76 11th RoboCup International Symposium, Atlanta, GA, JUL 09-10, 2007.

[7] Hayashi, Y., Fujiyoshi, H.: Mean-shift-based color tracking in illuminance change. In Visser, U And Ribeiro, F And Ohashi, T And Dellaert, F, ed.: Robocup 2007: Robot Soccer World Cup Xi. Volume 5001 of Lecture Notes In Computer Science., Heidelberger Platz 3, D-14197 Berlin, Germany, Springer-Verlag Berlin (2008) 302–311 11th RoboCup International Symposium, Atlanta, GA, JUL 09-10, 2007.

[8] Kashanipour, A., Milani, N.S., Kashanipour, A.R., Eghrary, H.H.: Robust color classification using fuzzy rule-based particle swarm optimization. In Li, D And Deng, G, ed.: Cisp 2008: First International Congress On Image And Signal Processing, Vol 2, Proceedings, 10662 Los Vaqueros Circle, Po Box 3014, Los Alamitos, Ca 90720-1264 Usa, Tianjin Univ Technol, Ieee Computer Soc (2008) 110–114 1St International Congress On Image And Signal Processing, Sanya, Peoples R China, May 27-30, 2008.

[9] Hildebrand, L., Fathi, M.: Knowledge-based fuzzy color processing. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **34**(4) (Nov. 2004) 499–505

[10] Reyes, N.H., Dadios, P.E.: Dynamic color object recognition using fuzzy logic. Journal of Advanced Computational Intelligence and Intelligent Informatics **8** (2004) 29–38

[11] Playne, D.P., Mehta, V.D., Reyes, N.H., Barczak, A.L.C.: Hybrid fuzzy colour processing and learning. In: ICONIP 2007, Part II, LNCS 4985. (2008) 386 – 395

[12] Browning, B., Veloso, M.: Real-time, adaptive color-based robot vision. In: Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on. (2005) 3871–3876

[13] Chalupa, L.M., Werner, J.S., eds.: The visual neurosciences. Volume 2. MIT Press (2004)

[14] Ebner, M.: Color Consancy. Wiley (2007)

[15] Swain, M., Ballard, D.: Indexing via color histograms. Computer Vision, 1990. Proceedings, Third International Conference on (Dec 1990) 390–393

[16] Heinemann, P., Sehnke, F., Streichert, F., Zell, A.: Towards a calibration-free robot: The act algorithm for automatic online color training. In Lakemeyer, G and Sklar, E and Sorrenti, DG and Takahashi, T, ed.: RoboCup 2006: Robot Soccer World Cup X. Volume 4434 of Lecture Notes In Artificial Intelligence., Heidelberger Platz 3, D-14197 Berlin, Germany, Springer-Verlag Berlin (2007) 363–370 10Th International Robocup Symposium, Bremen, Germany, Jun 19-20, 2006.

[17] Koschan, A., Abidi, M.: Digital Color Image Processing. Wiley-Interscience (2008)

[18] Gonzalez, R., Woods, R.: Digital Image Processing. 2nd edn. Prentice-Hall (2001)

[19] Holst, G.C.: CCD Arrays, cameras, and displays. JCD Publisishing and Spie Optical Engineering Press (1996)

[20] Yadid-Pecht, O., Etienne-Cummings, R., eds.: CMOS imagers : from phototransduction to image processing. Kluwer Academic (2004)

[21] Bayer, B.E.: ”color imaging array”. U. S. Patent 3,971,065, July, 20, 1976.

[22] Kim, D.Y., Park, H.K., Chung, M.J.: A robust and fast color-extracting using a look up table. In: Proceedings of the Fourth International Symposium on Artificial Life and Robotics. (1999) 650–653

[23] Stachowicz, M.S., Lemke, D.: Color recognition. In: Information Technology Interfaces, 2000. ITI 2000. Proceedings of the 22nd International Conference on. (2000) 329–334

[24] Kitano, H.: Research program of robocup. Applied Artificial Intelligence **12**(2-3) (1998) 117–125

[25] Kim, J.H., Seow, K.T.: Soccer Robotics. Springer (2004)

[26] Reyes, N.H., Barczak, A.L., Messom, C.: Fast colour classification for real-time colour object identification: Adaboost training of classifiers. In: In Proceedings of the Third International Conference on Automonous Robots and Agents. (2006) 611–616

[27] Jong, D., Alan, K.: An Analysis of the Behavior of a Class of Genetic Adaptive Systems. PhD thesis, University of Michigan (1975)

[28] Applications, N.: Os market share. http://marketshare.hitslink.com/os-market-share.aspx?qprid=9 [Online; accessed 18-February-2009].

Appendix A

Proposed System : FCCF Suite

The proposed system called *FCCF Suite* primarily offers users to evaluate FCCF with various parameter sets. The system designed for real-time vision processing with cross-platform capability. Many open-source type tools were incorporated into the proposed system to maximise user accessibility while focusing on the FCCF core logic. Furthermore, the system easily extends its functionality in real-time mobile robot applications. The system is mainly written in C++ with Qt extension for GUI functionality.

A.1 Licences

All of the incorporated libraries were based on open-source type licences. They are free to use and modify in non-commercial uses. Specific library documents can be found from the source code packages of each library.

A.2 Software Integration

A.2.1 Qt

Qt is a cross-platform application development framework, that is widely used for the development of GUI programs. Qt uses non-standard C++, extended by an additional pre-processor that generates the standard C++ code which is necessary to implement Qt's extensions on various platforms. It is produced by the Norwegian company Qt Software, formerly known as Trolltech, a wholly owned subsidiary of Nokia since June 17, 2008.
The Qt Open Source Edition is freely available for the development of Open Source software governed by the GNU General Public License versions 2 and 3.
QT is used in the proposed system as main GUI program as well as the timer provider for accessing the video camera and serial port.

A.2.2 OpenCV

OpenCV is a cross-platform computer vision library originally developed by Intel that runs on Windows, Mac OS X, Linux, PSP, VCRT (Real-Time OS on Smart camera) and other embedded devices. OpenCV is free for commercial and research use under a BSD license.
OpenCV is used as the image acquisition system.

A.2.3 QextSerialPort

QextSerialPort is a cross-platform serial port access library developed by Stefan Sander and fellow developers. QextSerialPort is subject to public domain license from SourceForge.net which offers GPL, LGPL, MPL, BSD License, Apache Software License and MIT License. However, the developer have not decided as of now.
QextSerialPort is used in the proposed system as the serial port controller which

accesses the RF signal board for controlling robots.

A.2.4 TinyXML

TinyXML is a C++ XML parser, capable of parsing an XML document. It builds from that a Document Object Model (DOM) that can be read, modified, and saved. TinyXML is originally developed by Lee Thomason and free for any purpose under zlib license.
TinyXML is used in the proposed system for colour classification file accessing.

A.2.5 GAlib

GAlib is a C++ Genetic Algorithm library developed by Matthew Wall at MIT. GAlib has the capability of adding evolutionary algorithm optimisation to almost any program using any data representation and standard or custom selection, crossover, mutation, scaling, and termination methods. GAlib is free to use in any purpose under BSD-style license with the following statement:
This research was performed using GAlib, a library of Genetic Algorithm components (http://lancet.mit.edu/ga/).
GAlib is used in the proposed system's Genetic Algorithm engine.

A.3 Features

A.3.1 FCCF

The proposed system fully implements the FCCF algorithm. It loads images from either file or video camera and applies FCCF algorithm with specific colour classification parameters and contrast rules on the entire image or specific target areas. The proposed system is also capable of generating optimal colour classification parameters and contrast rules from CCRE or Genetic Algorithm engine. It also supports loading, saving and manually editing any of the colour classification parameters and contrast rules.

Variable Colour Depth Look-Up Table

The proposed system fully supports the Variable Colour Depth algorithm and its corresponding look-up table with dynamic memory allocation.

Colour Contrast Rule Extraction (CCRE)

The proposed system is capable of extracting the best colour contrast rules for any given target colour, contrast angles and colour depth. The CCRE technique introduced in previous FCCF research [11].
Furthermore, the system extends CCRE to cover Variable Colour Depth in the search space.

Genetic Algorithm Engine

The proposed system is capable of extracting optimal colour classification parameters and contrast rules via a Genetic Algorithm.

A.3.2 Cross-Platform Compatibility

The proposed system has been tested under Windows, Mac OS X, Linux which covers more than 98%[28] of the operating systems currently used and compatible with almost all video cameras supported in these operating systems. The key advantage of having cross-platform compatibility is that the system does not require OS-specific software layers. Moreover, in the event that the target OS gets obsolete, there is no need to rewrite the system anymore as it could still be run in another OS. Another advantage is that the system could be developed by collaborating teams using different OS.

A.3.3 Video Capture

The proposed system utilises OpenCV to access video capture devices. However, it is worth-noting that many video image parameters, such as resolution, frame speed and colour space are dependent on the video capture device and its device driver software. Furthermore, the maximum frame speed may not be fully realised due to various host hardware factors, such as slow interface speed, high CPU utilisation, and slow video adapter.

A.3.4 Real-Time Object Tracking

The proposed system tracks multiple colour objects in real-time. The system analyses captured video images to identify candidate colour clusters. These colour clusters is then used for recognising the objects and tracking them. Target objects are defined in terms of a combination of colour patches, with predefined sizes. The proposed system adopts a search window technique that limits the object search space. When the target object is missing from the current search window, the system automatically enlarges the search window until the target object is found.

A.3.5 GUI System

The proposed system offers full GUI accessibility. It is capable of displaying real-time video images with object tracking information. Figure 76 shows a snapshot of the system.

A.3.6 Robot Control

The proposed system offers an interface to an RS232C serial port which is widely used in controlling RF-communication devices. This is particularly useful for interfacing with remote objects such as robots.

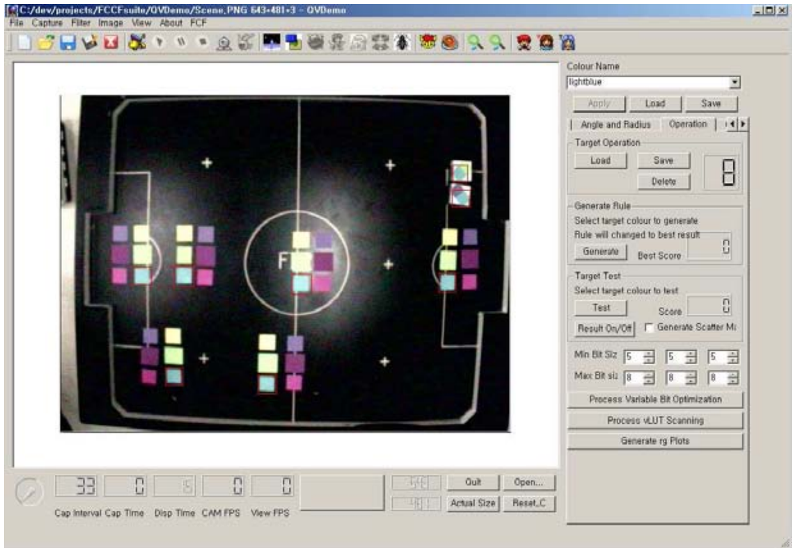

Figure 76: A Screen-Shot of the Proposed System.

A.4 Test-Bed Hardware Specifications

The proposed system has been developed and tested on an Intel Pentium 4 2.0 Ghz CPU with 1 GByte of memory, capable of processing 640*480 24-bit colour image at a rate of 30 frames per second. This is able to track 24 objects in different colours at the same time. The system is connected to an IEEE 1394 interface video camera, capable of capturing a 640*480 YUV 422 colour image at a rate of 30 frames per second.