

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

University of Alberta

ADAPTIVE COLOUR CLASSIFICATION FOR ROBOCUP WITH GAUSSIAN
MIXTURE MODEL

by

Xiaohu Lu



A thesis submitted to the Faculty of Graduate Studies and Research in partial
fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Spring 2005



Library and
Archives Canada

Bibliothèque et
Archives Canada

0-494-08134-1

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

ISBN:

Our file *Notre référence*

ISBN:

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work.

– John von Neumann

*To my wife Xiaoye,
and my son Jacob.*

Abstract

Colour has been used in many computer vision applications, such as image segmentation, object tracking and recognition. The appearance of an image is affected by illumination and so colour-based vision applications have often faced the problem of colours being sensitive to illumination variation. A static colour model can not handle illumination variation and so an adaptive colour model was introduced to deal with dynamic illumination. Our work is motivated by the need for colour classification in robocup research. We have developed an adaptive colour classification algorithm that uses a two component Gaussian Mixture Model (GMM) to model a colour distribution in YUV colour space. The components of this model represent the diffuse and the specular parts of the dichromatic reflectance model. The GMM is derived from classified colour pixels using the standard Expectation-Maximization (EM) algorithm, and the colour model is repeatedly updated with the derived GMM. We propose the novel idea that a GMM with two Gaussian components is an accurate and complete representation of the colour distribution of a dichromatic surface. This work is of practical significance because our adaptive system provides accurate colour classification under variant lighting conditions and it outperforms the previous colour vision system without adversely affecting efficiency.

Acknowledgements

I wish to express my gratitude to my supervisor, Dr. Hong Zhang. I am deeply thankful for his continuous support, advice, and for sharing his knowledge and insight.

I also wish to thank my colleagues in the robotics lab and the CIMS lab. Their advice and critiques helps to make this work happen.

Figure 1.1 and Figure 1.3 are used with the permission from the original author, Stephen Westland. Figure 1.2 is used with the permission from the original author, Helga Kolb. Figure 1.4 is generated with the code provided by Gernot Hoffmann, and used with the author's permission. I wish to thank these people for their help.

Table of Contents

1	Introduction	1
1.1	Background	1
1.1.1	What is colour?	1
1.1.2	The history of colour science	6
1.2	Motivation	8
1.2.1	How is colour used in computer vision?	8
1.2.2	How does illumination affect colour-based computer vision?	10
1.3	Approach	14
1.4	Structure of this thesis	14
2	Previous Work	16
2.1	Colour Constancy	17
2.1.1	Retinex Theory	18
2.1.2	Gamut Mapping	21
2.1.3	Colour by Correlation	25
2.1.4	Dichromatic Reflectance Model	27
2.2	Adaptive Colour Modeling	29
2.2.1	Non-Parametric Modeling	30
2.2.2	Parametric Modeling	33
2.3	Colour Vision in RoboCup	36
2.3.1	Summary	36
2.3.2	Vision system of previous teams	38
3	Methodology	45
3.1	Colour Modeling	46
3.2	Colour Space	61
3.3	Colour Adaptation	64
4	System Description	69
4.1	Off-line Colour Calibration	72
4.2	On-line Adaptive Colour Classification	75
5	Results	81
5.1	Adaptive vs Non-Adaptive	84
5.2	RGB vs YUV	86
5.3	Diffuse vs Specular and Diffuse	87
6	Discussion and Conclusion	101
	Bibliography	106

List of Figures

1.1	Electromagnetic radiation spectrum.	3
1.2	The physiological structure of human eye	4
1.3	1931 CIE tristimulus standard	5
1.4	1931 CIE chromaticity diagram	5
1.5	A simple scene with a single light source and no inter-reflection	11
2.1	Visualization of the first part of the gamut mapping procedure	24
2.2	Visualization of the second part of the gamut mapping procedure	24
3.1	A typical camera view of robocup team and the field. Pink, green, blue, and yellow blobs are illustrated with ×, +, circle, and square.	47
3.2	Colour distribution of the colour patches used on top of the robot in RGB space	50
3.3	Histogram of an object with dichromatic reflectance	51
3.4	The evolution of the distribution of green colour in RGB space	53
3.5	The evolution of the distribution of green colour in YUV space	54
3.6	The evolution of the distribution of green colour in UV space .	55
3.7	The colour distribution of a blue colour patch in RGB space .	65
3.8	The colour distribution of a blue colour patch in UV space . .	66
4.1	System GUI. Pink, green, and blue blobs are illustrated with ×, +, and circle.	71
4.2	Adhoc-colourpicker GUI calibration step 1. Pink, green, blue, and yellow blobs are illustrated with ×, +, circle, and square.	72
4.3	Adhoc-colourpicker GUI calibration step 2	73
4.4	Offline colour calibration	76
4.5	Initialized colour blob. Pink, green, and blue blobs are illustrated with ×, +, and circle.	77
4.6	Online adaptive colour classification	79
4.7	Colour classification using LUT built from initialized colours. The classified colour pixels are illustrated in white.	80
5.1	The correctly classified pixels with adaptive colour classification vs non-adaptive colour classification under gradually changing illumination from 70 Lux to 256 Lux.	89
5.2	The gradual illumination variation on the field from 70 Lux to 256 Lux.	90
5.3	The color variation under gradually changing illumination from 70 Lux to 322 Lux.	91
5.4	The color variation under gradually changing illumination simulated by changing the camera gain from 220 to 550.	92

5.5	Comparison of the number of correctly identified robots between adaptive colour classification and non-adaptive colour classification under gradually increasing lighting. The total number of robots is 4.	93
5.6	Comparison of the classification accuracy between adaptive colour classification and non-adaptive colour classification under gradually changing lighting.	94
5.7	Comparison of the correctly classified pixels between adaptive colour classification and non-adaptive colour classification under gradually changing lighting.	95
5.8	Comparison of the correctly classified colour blobs between adaptive colour classification and non-adaptive colour classification under gradually changing lighting.	96
5.9	Comparison of the correctly identified robots by adaptive colour classification under gradually changing lighting between RGB and YUV colour space	97
5.10	Comparison of the classification accuracy of adaptive colour classification under gradually changing lighting between RGB and YUV colour space.	98
5.11	Comparison of the correctly identified robots by adaptive colour classification in YUV space under gradually changing lighting between using only the diffuse component and both diffuse and specular components	99
5.12	Comparison of the classification rate between using only diffuse component and using both diffuse and specular components with adaptive colour system under gradually changing lighting.	100

Chapter 1

Introduction

1.1 Background

1.1.1 What is colour?

Colour is a phenomenon that we humans are familiar with and it has played an important role in our everyday life through thousands of years of the human history. Colour is not only used in decoration and art, from traffic lights to life jacket to the clothes we wear, colours are all around us and they affect our lives in a great manner.

What is colour? It is interesting to know that not many of us can answer this question even though colour is such a common phenomenon. As a matter of fact, even many colour science literatures lack of definition of colour despite the fact that they study colours. The main reason why it is hard to define colour is that colour is not part of the object property, rather it is the property of the human sensation. An object does not have a colour, it appears to be of a certain colour only because the light this object gives off has a certain spectral power distribution which is perceived by human observer as a certain colour. The colour we perceive depends on many factors, such as the object material, the illumination condition, the observer, and etc. Even if all the physical conditions are well controlled to be invariant, there exists

a slight difference in colour sensation from one individual to another. In fact even the same individual has slightly different colour sensation from time to time. The subjectivity existing in colour perception makes an objective definition of colour almost impossible for a long time till people came up with a measurement of colours.

According to the Webster's Dictionary in 1913, colour is defined as a property depending on the relations of light to the eye, by which individual and specific differences in the hues and tints of objects are apprehended in vision. According to a textbook definition [62], colour is the characteristic of a visible object or light source by which an observer may distinguish differences between two structure-free fields of the same size and shape, such may be caused by differences in the spectral composition of the light concerned in the observation. In other words, colour is the perception by which we can tell two objects apart, when they have otherwise similar attributes of shape, size, texture, etc..

To understand what is colour, it is necessary to understand the physical property of colour and the physiology mechanism of human colour vision system. Colour is our perception of a light with a unique spectral distribution. Light is the electromagnetic radiation which consists of a mixture of radiations of different wavelengths and intensities. Our eyes are only sensitive to the electromagnetic radiation with wavelength in the range between 380 nm and 740 nm , which is called visible light. The spectrum of the light is the distribution of intensity at different wavelength. The light with different spectrum is perceived by human as different colour.

When we talk about what is colour, it is unavoidable to mention how we perceive colour. There are three types of cone cells in human vision system as colour receptors. One type is most responsive to blue light with wavelength

electromagnetic spectrum

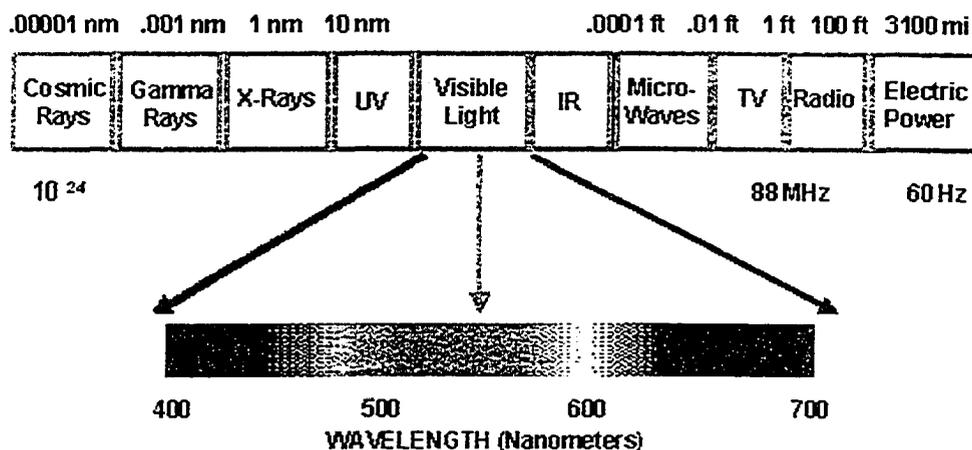


Figure 1.1: Electromagnetic radiation spectrum.

around 420 *nm*, and this type is sometimes called short-wavelength cones, S cones, or blue cones. The other two types are both responsive most to greenish light. One is most sensitive to yellowish-green light with wavelength around 564 *nm*, and this type is sometimes called long-wavelength cones, L cones, or red cones. The other is most sensitive to green light with wavelength around 534 *nm*, and this type is sometimes called middle-wavelength cones, M cones, or green cones. The sensitivity curves of the three types of cones are all roughly bell-shaped and overlap considerably. The incoming signal spectrum is thus reduced by the eye to three values, representing the intensity of the response of each of these types of colour receptors. The signal from the colour receptors is analyzed by nerve cells to calculate the colour of the light which reaches the retina, and then the signal is sent to the brain via visual pathway. Specifically, the signals from colour receptors are analyzed by

ganglionic cells which code colour information, and then the signals are sent to the parvocellular cell layers of the dorsal lateral geniculate nucleus (LGN) in the thalamus, onwards to the retinotopic mapped layers of the LGN, which are named as V1, V2, and so on. It is in fact our brain that perceives the colour. A number of clinical studies present evidence that without the development of the visual cortex inside the brain, people have a problem in colour perception even though they have completely normal visual system from retina to the pathway to brain.

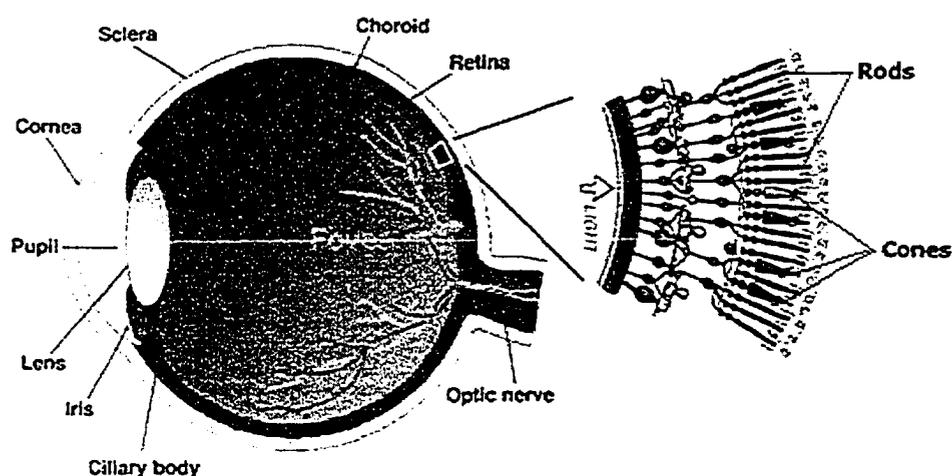


Figure 1.2: The physiological structure of human eye

Brightness, hue, and saturation are the three commonly used characteristics to distinguish one colour from another [18]. Brightness embodies the achromatic notion of intensity, it is in fact a subjective descriptor that is practically impossible to measure. Hue is an attribute associated with the dominant wavelength in a mixture of light waves, and it represents dominant colour as perceived by an observer. Saturation refers to the relative purity or the amount of white light mixed with a hue. The degree of saturation is inversely proportional to the amount of white light added. When we call an object red, green, or yellow, we are specifying its hue. When we call an object

CIE 1931 - 2° standard observer

Tristimulus values of the spectral colours

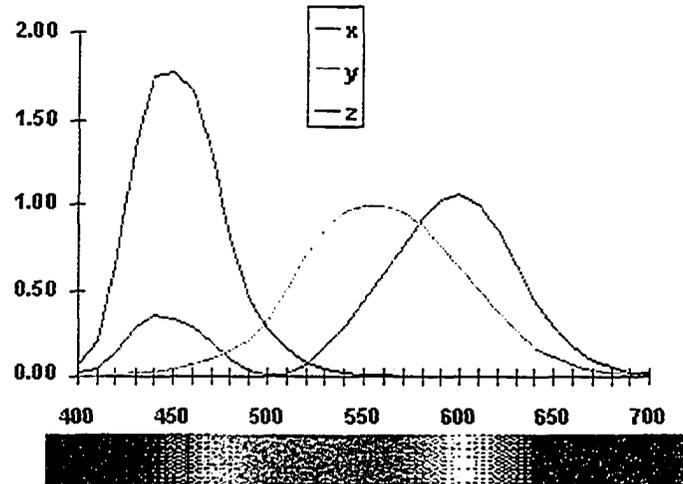


Figure 1.3: 1931 CIE tristimulus standard

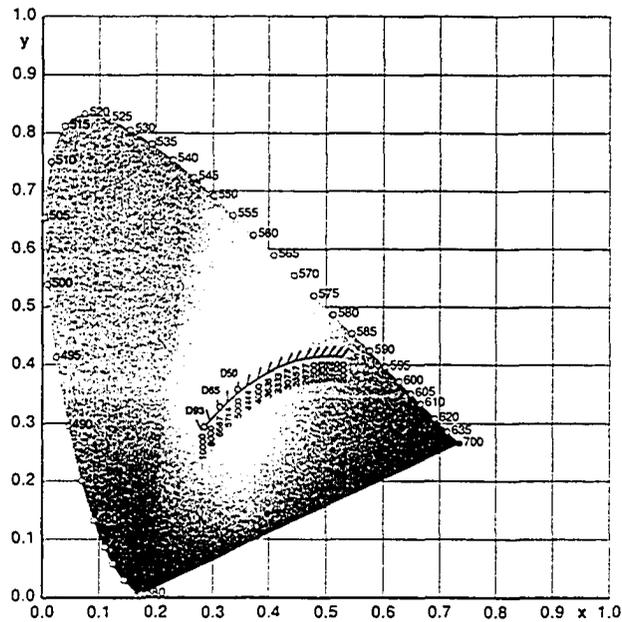


Figure 1.4: 1931 CIE chromaticity diagram

deep blue or light blue, we are specifying its saturation. Hue and saturation together are called chromaticity; therefore a colour is defined by its brightness and chromaticity [18]. The standard designated by Commission Internationale de l'Eclairage in 1931 uses three primary colours to represent a colour: blue with wavelength of $435.8nm$, green with wavelength of $546.1 nm$, and red with wavelength of $700 nm$. The amounts of red, green, and blue which form a colour are called tristimulus values and are denoted as X , Y , and Z . This notation is called the CIE-XYZ colour space. With this notation, a colour can be specified by its trichromatic coefficients, defined as [18]:

$$x = \frac{X}{X + Y + Z} \quad (1.1)$$

$$y = \frac{Y}{X + Y + Z} \quad (1.2)$$

$$z = \frac{Z}{X + Y + Z} \quad (1.3)$$

Based on the CIE-XYZ space notation, a colour can also be specified with the CIE chromaticity diagram. Because $x + y + z = 1$, CIE chromaticity diagram uses x and y to represent the colour composition, x represents red, y represents green, as shown in Figure 1.4.

1.1.2 The history of colour science

Although humans are always interested in the colour phenomenon through our recorded history, and lots of brilliant philosophers and scientists, from Plato to Aristotle to Leonardo da Vinci, all have different theories about colour, it was only in the seventeenth century when people started to have an understanding of the physics nature of colour through scientific studies.

In 1666 Sir Isaac Newton presented that white light is a mixture of all the colours of the visible spectrum. Using his famous prism experiments, Newton showed that when the white light passes through a prism, the light is separated into seven-colour band, and when the seven-colour band passes through a reversed prism, the colours recombine to form white light. Based on the experiments, Newton developed the Newton's colour circle which presents the additive mixing properties of colours.

In 1802 Thomas Young suggested that the human eye has three different kinds of colour receptors, and each kind of the receptors corresponds to the red, green, and blue primary colours. Hermann von Helmholtz provided more quantitative evidence on Young's idea; therefore this theory is called the Young-Holmholtz theory. It was not until about 1965 that the detailed physiological experiments were performed to measure the absorption of the different types of cones in the eye. Those experiments verified the Young-Holmholtz theory that there are indeed three types of cones.

A method of measuring colours is a necessity for the purpose of studying colour. The work done by James Clerk Maxwell in the 1860s could be considered as the basis for modern colourimetry, which is the science of measuring colours despite the fact that we perceive colours slightly differently. Maxwell explored the use of three primary colours and studied the problem of if additive combination of three primary colours can cover the entire range, also called gamut, of perceivable colours. He showed that although the three primary colours are not sufficient to cover the entire gamut of perceivable colours, the spectral primaries more widely separated in wavelength could be used to produce a wider range of perceived colours, and indeed the entire gamut of perceived colours could be covered. Also Maxwell recognized that the chromaticity (hue and saturation) of a coloured surface is relatively insensitive to

the brightness.

Detailed experiments carried out in the 1920s showed that the RGB primaries could indeed match all visual colours within a certain range, but that they could not match all the spectral colours, particularly in the green range. It was found that if a certain amount of red light was added to the colour being matched, then all colours could be matched. The quantitative results were expressed in terms of tristimulus values for the RGB primaries, but it was necessary to allow negative values for the red tristimulus values in order to match all colours.

In 1931 the Commission Internationale de l'Eclairage (CIE) defined a standard system in which all the tristimulus values would be positive and in which all visible colours could be unambiguously represented by two chromaticity coordinates x,y . Mapping the visual colours led to the now familiar horseshoe curve in the x,y plane known as the CIE chromaticity diagram (see Figure 1.4). It is the basis for most quantitative colour measurement at present. It is noted that the 1931 CIE standard chromaticity diagram has a problem that the distance between two points on the diagram is not proportional to the perceived colour difference. In 1976 a new CIE standard was released to correct this problem, but it did not gain acceptance. The 1931 CIE standard is still universally used.

1.2 Motivation

1.2.1 How is colour used in computer vision?

Colour started to play a role in computer vision not long ago, as the colour cameras and colour digitizers have become more and more common during the past twenty years. Before that, computer vision mostly dealt with gray-scale

images, which contain only the information of the image intensity. With the development of colour devices, the application of colour in computer vision has progressed greatly in many different areas, such as image retrieval, image synthesize, image segmentation, object recognition, object tracking, and etc..

Colour has been used as an important image feature in computer vision applications because of its advantages over other image features. First of all, colour can be a powerful image feature that simplifies object identification and extraction from a scene [18]. Colour has the benefit of being fairly invariant with respect to change in the object geometry caused by object moving in space, such as its rotation, translation, or scaling. This is particularly important when we are interested in detecting an object with known colour appearance but unknown geometry or variant geometric properties. In this case, a simple colour segmentation could separate the object from the background while other image features might fail to handle the variation in the feature space. For example, when we detect a football player who is playing a game from the image of the football field, using colour would be more reliable than using shape or contour of the player, as the colour information of the player keeps more invariant than its shape or contour during his movement. Secondly, colour has the benefit of being computationally efficient as a low-level image feature, especially when compared with high-level image features such as object edge, shape, or contour. For example, a bottom-up colour-based tracking algorithm can locate an object by first labeling each pixel in the image into one of the colour classes and then applying spatial constraints, while a contour-based tracking algorithm might suffer from inefficiency as it involves searching in a high-dimensional feature space. This makes colour a suitable candidate for computer vision applications with high efficiency requirement, such as real-time object tracking and object recognition.

There is a large amount of research work in colour-based human skin recognition [59, 64, 10, 19, 23, 37, 40, 65, 55, 66, 52, 53, 26, 61]. Tracking and detecting human skin colour from video sequence serves as an effective approach in applications, such as video surveillance, face detection, human-computer interaction, and etc. Colour has also been used in automatic vehicle navigation, robot navigation, object tracking, and etc.

The motivation of this thesis is one typical example of colour vision application, the small size league of the RoboCup competition. In robocup competition, as two teams of robots play soccer, the vision system for each team serves as the “eye” of the team to find the locations and the identities of the robots on the field, before the information is processed for making game strategy. This requires the vision system to be both accurate and efficient, as it is essential that all the robots and the ball be located and identified in real-time in order to effectively play the game. Each robot has a colour patch of its own team colour on the top, and each team uses a certain colour patch pattern to identify its own team members. These colour patches serve as markers of the player so that the vision system can easily locate and identify each robot. In the past this approach has been successfully used in robocup competition. Figure 3.1 shows a typical view of the field with robots and ball from the camera.

1.2.2 How does illumination affect colour-based computer vision?

With the benefits of the colour in computer vision being mentioned above, we should also note that colour faces the inevitable problem of being affected by illumination. It is commonly acknowledged that illumination changes the appearance of an image taken from a fixed scene. Generally illumination con-

ditions can have a critical impact on a computer vision application. This is also true for colour-based vision applications. As the image appearance changes due to the illumination, it leads to difficulties of stability, robustness, and accuracy of the colour-based vision. For example, when a person walks into shade, the light casting on his face goes through a dramatic change, which leads to the change of his skin colour. In this situation, a face detection application relying on the skin colour under light will have difficulty in locating the face in shade. This issue has been addressed by many researchers, and intense research work has been devoted trying to solve the problem associated with illumination.

We will briefly reemphasize this issue here by addressing the imaging formation problem.

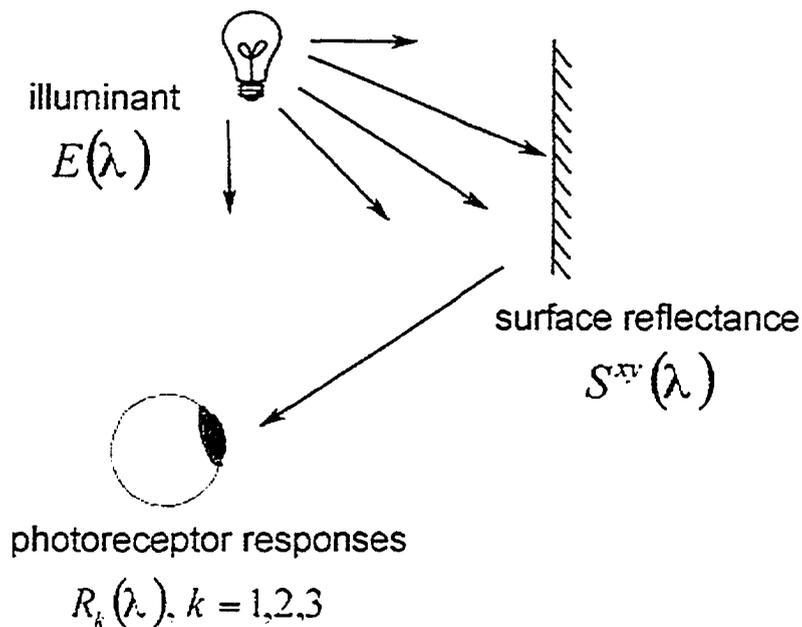


Figure 1.5: A simple scene with a single light source and no inter-reflection

We can treat a digital image as a sampling of the light signal which can

be modeled by a continuous function of wavelength and geometric variables. For a colour image, we have three samples overlapping at the same spatial location. Figure 1.5 illustrates a simple scene in which there is only a single light source, and the light reaches the imaging device after being absorbed and reemitted by just one surface [36]. The camera response at each location xy in the image can be expressed by the equation

$$\rho_k^{xy} = \int E(\lambda)S^{xy}(\lambda)R_k(\lambda)d\lambda, k = 1, 2, 3 \quad (1.4)$$

where $S^{xy}(\lambda)$ is the surface spectral reflectance function of a surface patch imaged on location xy , $E(\lambda)$ is the spectral power distribution of the light incident on the surface patch, and $R_k(\lambda)$, $k = 1, 2, 3$ are spectral sensitivities of the imaging sensor, or photoreceptor in case of human vision. These functions are all indexed by wavelength λ in the electromagnetic spectrum. A more realistic model of a scene would include the possibility of multiple light sources and inter-reflections between the surfaces, and would also take into account the orientation of the surfaces. In both the simple and the realistic models, the sensor response on each channel depends on the spectral properties of both the illuminant and the surfaces present in a scene.

This model has been verified as being adequate for computer vision over a wide variety of systems. It is also assumed for the human visual system, and forms the basis for the CIE colourimetry standard. Here, ρ_k^{xy} are the X, Y and Z colour coordinates. In the common case of three camera channels, ρ_1^{xy} is the linearized red channel, designated by R, ρ_2^{xy} is the green channel, designated by G, and ρ_3^{xy} is the blue channel, designated by B.

Formula 1.4 illustrates that an image depends not only on the physical content of the scene, but also on the illumination incident on the scene, and the characteristics of the camera. This phenomenon could lead to a problem of

computer vision applications in which we are interested in the physical content of the scene. If the illumination change affects object colours beyond the tolerance of the vision system, this change of the illumination can cause significant problems. This is particularly a problem for an outdoor video surveillance or face recognition system, which lacks of a well controlled lighting environment. Usually a static colour model can not capture the wide distribution of this colour due to dynamic lighting condition, and the accuracy of the system is affected by the illumination change in the environment. For the robocup competition, fortunately indoor lighting usually does not have dramatic change during the game; therefore illumination has not been a serious threat to the vision system. However, as the latest regulation starts to enforce less controlled lighting on the field, we believe it is the trend that in the future robot soccer game will be played on a field with ambient lighting, and the vision system will need to be less constrained by the lighting condition. A vision system which can classify colour under dynamic lighting will be considered in robocup as a necessity. Besides the robocup, illumination effect is a very common problem in most of the computer vision applications. It would be of significant importance to be able to deal with illumination change for a general vision application.

There are different approaches in dealing with the effect of the illumination on the image. One approach is colour constancy, which attempts to recover the intrinsic physical property of a scene from images. Another approach is adaptive colour modeling, which attempts to model the evolution of the colour due to lighting change so that a colour model can adapt to different illuminations. Both approaches will be discussed in detail in Chapter 2 as previous work. Adaptive colour modeling is the focus of this study.

1.3 Approach

How to deal with the illumination effect on the colour has attracted attention from many researchers. It is noted that some colour spaces have the property of being less sensitive to lighting effect, such as HSV or YUV space, so choosing a proper colour space could compensate the lighting variation to some extent. Unfortunately such a colour space can only deal with mild lighting change; therefore they do not always serve as a reliable solution.

In this work we present an algorithm for colour classification using adaptive colour modeling, and we present a colour vision system for real-time colour classification in robocup which implements our algorithm. We use Gaussian Mixture model (GMM) of two components in YUV space to model the colour distribution of each colour, according to the dichromatic reflectance model. By using the GMM to evaluate the probability of a pixel belonging to a certain colour class, and then labeling this pixel after thresholding its probability, an accurate colour classification of this colour from an image can be achieved. The GMM is adapted with an exponentially decaying function as the lighting changes so that it handles dynamic lighting condition efficiently. The experiments demonstrate that the colour classification system implemented with our algorithm remains accurate, robust, and efficient under dynamic lighting.

1.4 Structure of this thesis

This thesis presents a study of adaptive colour classification. Chapter 1 presents an introduction of the colour science and the motivation of this study. In Chapter 2 previous work on colour vision studies is described with focus on three aspects: colour constancy, adaptive colour modeling, and colour vision in robocup domain. Chapter 3 presents our algorithm, and chapter 4 presents the

description of colour vision system implemented with our algorithm. Chapter 5 presents our experimental results on the robustness and the accuracy of our algorithm. As the last chapter of this thesis. Chapter 6 presents a discussion of our algorithm and the conclusion.

Chapter 2

Previous Work

The research on how to deal with the effect of illumination on image appearance has been following two approaches based on two different schools of philosophy. One approach is colour constancy, which refers to the ability of a vision system to diminish or even remove the effect of the illumination such that it can understand the physical scene more precisely. Colour constancy originally comes from the colour perceptual constancy of human vision. Although there is ample evidence that the human vision system exhibits some degree of colour constancy [39, 4, 35], its mechanism is still not clear. It is commonly acknowledged that estimating illumination from images is an ill-posed problem, and most existing colour constancy algorithms rely on assumptions of reflectance models and illumination models to simplify the problem. As these colour constancy algorithms are usually accompanied with strong assumptions and heavy computational burden, it is not feasible to use these algorithms in a real-time application. The second approach is adaptive colour modeling. This approach does not attempt to recover the illumination-independent description of a scene's surface colours, instead it tries to model the evolution of the colour under different lighting conditions. The focus of the adaptive colour model algorithms is to find a more suitable colour model and to adapt this colour model as lighting change. Adaptive colour model has attracted intense

research interests in the past, and its use in colour tracking applications shows promising results [60, 63, 42].

In this chapter, we will introduce the previous work on how to compensate the effect of illumination on image appearance. Firstly, we will describe the studies previously done in this area from two aspects, colour constancy and adaptive colour modeling. Secondly, we will describe the different approaches used in colour vision of robocup and how they manage to handle illumination change.

2.1 Colour Constancy

Colour constancy is a term originated from the phenomenon observed in human colour perception. Colour constancy is a feature of human colour perception system which ensures that human's perception of colours of objects remains constant even though there is dynamic illumination. Humans have a fairly good ability to keep a constant colour perception under varying lighting conditions. We have been aware of this phenomenon long time ago, although the mechanism behind it is still not well understood. For example, Thomas Young had noted in 1807 that white paper still looks white even when illuminated by the red light from a fire. Colour constancy helps us to easily recognize objects under varying lighting, and this is of great importance for us to survive in the world of changing lighting.

It is believed by some researchers that human colour perception system takes the colour information of the scene, calculates the ratio of different colour components perceived by different types of colour receptors, and uses the information of ratio between different colour components to approximate the illumination condition [30]. The approximated illumination condition helps us to discount the illumination effect on images and regain the actual colour

information of the objects. The algorithm behind this mechanism is not clear yet.

Colour constancy is a desired visual performance in computer vision. As mentioned before, illumination has significant influence on image appearance and it leads to difficulties for computer vision to recognize a colour under varying lighting. It would greatly help computer vision to deal with varying illumination if it has the ability of colour constancy. Lots of research works have been carried out for the goal of achieving colour constancy in computer vision during the past decades, and most of them have attempted to base their ideas on the understanding of how human vision system maintains colour constancy. The following sections describe several major algorithms in colour constancy research area.

2.1.1 Retinex Theory

Retinex theory proposed by Edwins Land in 1977 has been considered as one of the most important theories on colour perceptual constancy, as it was the first attempt at developing a computational model for human colour constancy [5]. The name retinex comes from retina and cortex, which are two important physiological structures of human visual system. Retinex theory was originally proposed to model human vision, and since then it has been greatly extended to colour constancy research.

In the late 1950s, Land reported some interesting results from his experiments on colour perceptual constancy. These results can not be explained with the classic colour vision theory. In fact, these experiments led to the development of the retinex theory and the retinex algorithm.

In one of his early experiments, Land used black-and-white transparency film to photograph a highly coloured scene. As he photographed this scene,

he had a red filter in front of the camera lens to take the first photo, also he had a green filter in front of the camera lens to take the second photo. Now these two black-and-white photos were almost identical, except that the intensity of the image was slightly different between these two photos. Land then projected these two photos with two projectors and superimposed them on a screen, with a red filter in front of the projector which projected the photo taken with a red filter, and a green filter in front of the projector which projected the photo taken with a green filter. Surprisingly, the superimposed projections on the screen became a full colour image. This phenomenon can not be explained by classic colour vision theory. According to the traditional colour mixture theory, the red and the green can only mix into yellow with different shades, but not full colours.

The experiment described above together with other experiments confirm the theory that the visual system converts the light into three sets of lightness values independently with each of the three types of colour receptors. Also these experiments imply that human perceives colours not by measuring the absolute value of the wavelength of the light, instead that colour is perceived by calculating the ratio between the longer and the shorter wavelength of light. Based on these experiments, Land proposed that the information people require to see colour is not in the wavelength of light reflected by an object, instead that colour information is encoded in the ratio of the longer and the shorter wavelengths of light reflected by an object.

The purpose of the retinex algorithm is to compute lightness values which are invariant under changes of illumination, just like human performance is roughly invariant under similar changes. At each pixel x , the lightness values depend only on the surface reflectance $S^x(\lambda_n)$ at this pixel, and not on the spectral power distribution of the light or the surface reflectance functions of

the other surfaces in the scene [5].

Retinex algorithm estimates the lightness of a surface in each channel by comparing the quantum captured at each pixel to the value of some statistics—originally the maximum— found by looking at a large area around the pixel or photoreceptor. The ratios of these quantities, or the differences of their logarithms, are the descriptors of interest. The original Retinex algorithm is to follow random paths from the pixels of interest [38, 30]. As each path is followed, the ratio, or the difference of the logarithms, of the response in each channel for adjacent pixels is computed. If we start from a pixel x_1 and follow a random path, we then randomly select a neighboring pixel x_2 . The difference of the logarithms of the sensor response at these two pixels is calculated and stored such that

$$\begin{aligned} A(x_2) &= A(x_2) + \log(\rho^{x_2}) - \log(\rho^{x_1}) \\ N(x_2) &= N(x_2) + 1 \end{aligned} \quad (2.1)$$

By following this random path, a number of pixels along this path are processed in the same manner, such that for a pixel x_i ,

$$\begin{aligned} A(x_i) &= A(x_i) + \log(\rho^{x_2}) - \log(\rho^{x_1}) \\ N(x_i) &= N(x_i) + 1 \end{aligned} \quad (2.2)$$

$A(x_i)$ and $N(x_i)$ are both initialized to zero. $A(x_i)$ stores the updated value at pixel x_i contributed by the difference between pixel x_i and the starting point of the path x_1 . $N(x_i)$ stores the frequency of how many times pixel x_i has been counted along all paths. After processing a number of random paths starting from a number of pixels of interests, the lightness value at pixel x_i is calculated by

$$l_{x_i} = \frac{A(x_i)}{N(x_i)} \quad (2.3)$$

Retinex based methods assume that small spatial changes in responses are due to changes in the illumination and large changes are due to surfaces changes. If the ratio is close to one, it is assumed that the difference is due to noise or varying illumination, and the ratio is treated as exactly one. In this case, the algorithm looks like this

$$\begin{aligned} A(x_i) &= A(x_i) \\ N(x_i) &= N(x_i) + 1 \end{aligned} \tag{2.4}$$

If the ratio is sufficiently different from one, then it is used as is. The ratios are then combined to determine the ratio of the response of the pixel of interest to the largest response found in the path. Finally, the results for all the paths are averaged.

In summary, with the logarithmic representation, the essence of the algorithm is differentiation (to identify the jumps), followed by thresholding (to separate reflectance from illumination), followed by integration (to recover lightness).

While retinex theory challenges the classic colour vision theory, it is also challenged by many researchers [5]. Brainard argued that retinex theory is essentially equivalent to image normalization, and it corrects colour in a manner that depends strongly on the surfaces in the scene [5]. Based on this argument, Brainard claimed that retinex is not a colour constancy algorithm [5]. Despite of these arguments, retinex theory has been extended and used widely in computer vision area. For example, NASA has been using retinex theory for photo enhancement.

2.1.2 Gamut Mapping

The goal of colour constancy is to diminish the influence of the illumination on an image in order to get a more accurate description of the physical prop-

erty of the scene. The generic approach is to find the illuminant-independent description of the scene. An alternative approach is to recover the image appearance under a known illumination for an image taken under an unknown illuminant. This latter approach also achieves the goal of diminishing the effect of unknown illumination on an image to get a better understanding of the physical property of the scene. Gamut mapping is an approach which constrains the set of possible mappings from the images of the scene under the unknown illuminant to the image of the scene under the known canonical illuminant. It was presented by Forsyth and modified by Finlayson [16, 13].

Suppose we have a set of images $\{I_i, i = 1 \dots n\}$ of the scene, each of them taken under a unique unknown illuminant from the set of illuminants $\{L_i, i = 1 \dots n\}$, and we want to exam these images to see if they are all taken from the same scene. If we are given a known canonical illumination L' , and we are able to recover the image appearance of I_i to I'_i as if it is taken under illumination L' , then we can answer the question if these images are taken from the same scene easily through a straightforward comparison of the images I'_i . The image comparison is now an easy problem since it does not have to consider the influence of the different illumination.

Suppose we specify a known canonical illuminant, and all possible image pixels due to all known surface reflectance under this canonical illuminant compose a convex set, which is called a gamut. In this case it is called the canonical gamut. If we change the illuminant to an unknown one, all possible image pixels under this unknown illuminant also compose a convex set, which is called the unknown gamut. Assume the illumination change follows a diagonal model, these two gamuts also follow a diagonal transformation. If we are able to find the diagonal model of gamut transformation, we can derive the image appearance from unknown illuminant to known canonical illuminant.

Now the question is that since we only know the canonical gamut but not the unknown gamut, how do we calculate the diagonal transformation between these two gamuts? Fortunately, the image samples taken under unknown illuminant provide a subset of the unknown gamut, which is called the measured gamut. Gamut mapping tries to find the mapping between the canonical gamut and the unknown gamut using the measured gamut.

Figure 2.1 shows a visualization of the idea of gamut mapping. The triangle “abc” represents the convex hull which is the measured gamut, and the triangle “ABC” represents the convex hull which is the canonical gamut. As “abc” is the subset of the unknown gamut, the mapping from point a to “ABC” must be a point inside “ABC”. Although we don’t know where exactly this mapped point is, we can bound this mapping inside the convex hull of maps “aA”, “aB”, and “aC”. Now if we only consider point a, the set of all possible mapping is constrained to within this set.

Now if we repeat this process on more points of “abc”, we can get a different set of possible mappings from those points. The intersection of these different set of mappings is the constrained possible mappings from the unknown gamut to the canonical gamut. Figure 2.2 illustrates this process.

Finlayson modified the gamut mapping approach by reducing the dimensionality of the problem from 3D RGB space to 2D chromaticity space[13]. He showed that a 2D chromaticity space in the form of (R/B, G/B) can maintain the convexity of the gamut. By changing the problem to chromaticity space, the algorithm is robust with respect to the image difference caused by intensity variation, such as shadow or nonuniform lighting.

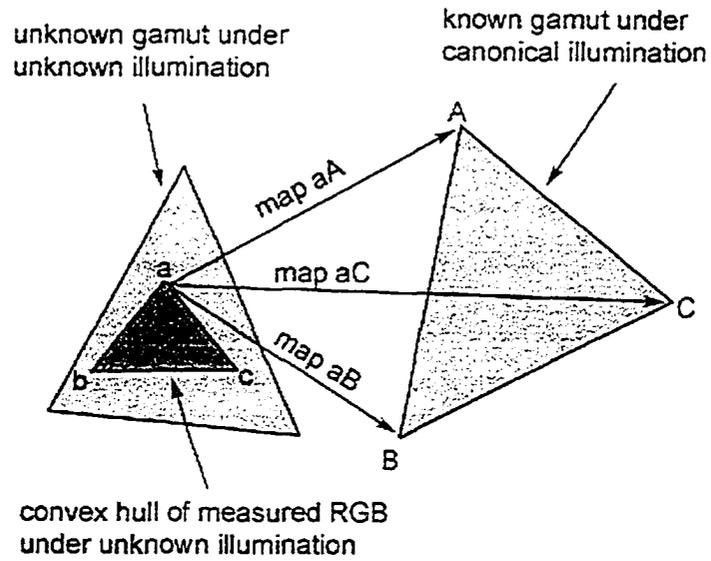


Figure 2.1: Visualization of the first part of the gamut mapping procedure

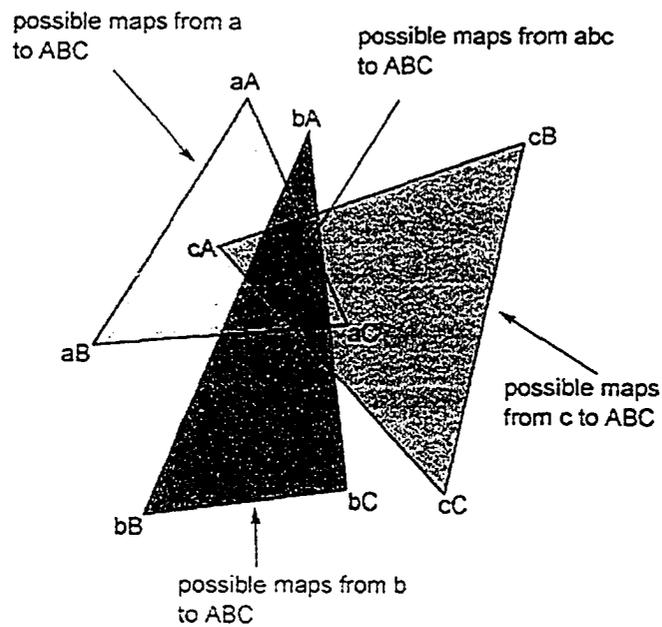


Figure 2.2: Visualization of the second part of the gamut mapping procedure

2.1.3 Colour by Correlation

Colour by correlation method is to solve the illuminant estimation problem: given an image taken under an unknown illuminant, how can we recover an estimation of the illuminant [14]? Colour by correlation method is presented by Finlayson et al. [15, 14]. The basic idea of colour by correlation is to pre-compute a correlation matrix which describes how compatible proposed illuminants are with the occurrence of image chromaticities. Each column in the matrix corresponds to a different training illuminant. Each row corresponds to possible chromaticity ranges resulting from a discretization of (r,g) space, ordered in any convenient manner.

There are two different versions of colour by correlation [15]. The first version calculates the elements of the correlation matrix under a given illuminant as follows. First, the (r, g) chromaticities of the reflectance in the training set under the given illuminant are computed using the camera sensors input. Then the convex hull of these chromaticities is found, and all chromaticities within this convex hull are identified as being compatible with the given illuminant. Finally, all entries in the column for that given illuminant corresponding to compatible chromaticities are set to one, and all other elements are set to zero.

Given an image taken under unknown illuminant, this correlation matrix can be used to estimate the illuminant chromaticity. First, the chromaticities in the image are converted to a vector which contains elements corresponding to the order of (r,g) used in the correlation matrix. An element is one if its corresponding chromaticity exists in the image, or zero otherwise. Multiplying the correlation matrix with this vector gives the correlation between the image and each of the illuminants in the correlation matrix. Ideally, the actual illuminant which the image is taken under has the maximal correlation with

the given image. As the possible illuminant might not be the only one, multiple illuminants which are close to the maximum are chosen as candidates, and their average is taken as the estimated illuminant.

The second version of colour by correlation algorithm introduces the probability of the chromaticities under a certain illuminant, and it recovers a measure of the likelihood that each of a set of possible illuminants is the scene illuminant. Using this likelihood information, an estimation of the scene illuminant can be recovered [14].

In this version, each entry of the correlation matrix is not one or zero; instead it stores the logarithm of the probability of a chromaticity under an illuminant. Now the correlation matrix can be used to compute the logarithm of the posterior distribution, which is the probability of the illuminant given an observation image, by multiplying the vector calculated from the image. This posterior distribution is calculated by applying Bayes's rule. The probability that the scene illuminant is I , given a set of observed chromaticities C , is given by Bayes's rule:

$$P(I|C) = \frac{P(C|I)P(I)}{P(C)} \quad (2.5)$$

If we assume uniform distribution for I , and since $P(C)$ is a normalization, we have:

$$P(I|C) \propto P(C|I) \quad (2.6)$$

Assume the independence of the observed chromaticities, we have:

$$P(C|I) = \prod_{c \in C} P(c|I) \quad (2.7)$$

Taking logarithms on it gives:

$$\log(P(C|I)) = \sum_{c \in \mathcal{C}} \log(P(c|I)) \quad (2.8)$$

This formula gives the results of multiplying the correlation matrix to the vector of the chromaticity observation. Each element of the resulting vector is the logarithm of the posterior probability under that illuminant.

2.1.4 Dichromatic Reflectance Model

Most reflectance models make the assumption that the object surfaces are Lambertian; i.e, they are perfectly matte and appear equally bright from all directions. This is a rather simplified model of the material's reflectance property as in practice most of the materials are not perfectly Lambertian. In fact, dielectric materials such as paper, paint, plastic, etc. are very common in the real world, and most of the dielectric materials are inhomogeneous and not perfectly Lambertian. Dielectric materials appear more or less glossy depending on the view angle and the lighting intensity, and this phenomenon can not be explained by the Lambertian model. The dichromatic reflectance model was first presented by Shafer in 1985 to explain the reflectance property of objects with surface of dielectric materials.

According to the dichromatic reflectance model, the light reflected from a surface of dielectric material usually consists of two distinct colours that correspond to the highlight areas and the matte areas. As dielectric material consists of a medium and some embedded pigment, when light hits the surface, the inhomogeneity of the surface material causes the light to partially reflect back into the air, and partially refract into the material body. The light reflected from the surface into the air is called surface reflection, and it usually appears as a highlight on the object. Depending on the roughness of the surface, the surface reflection can be in one direction or many directions. The

light refracted into the material body travels through the medium, refracted between the pigments, and eventually leaves the body. This portion of the light is called the body reflectance; it usually appears as the diffuse on the object. The body reflectance depends on many factors, such as the physics property of the medium, the size, the shape and the concentration of the pigments in the body.

The dichromatic reflectance model describes the reflected light from a dielectric material surface $L(\lambda, i, e, g)$, as the mixture of two portions, the surface reflection $L_s(\lambda, i, e, g)$ and the body reflection $L_b(\lambda, i, e, g)$.

$$L(\lambda, i, e, g) = L_s(\lambda, i, e, g) + L_b(\lambda, i, e, g) \quad (2.9)$$

where i , e , and g describe the angles of the incident and emitted light and phase angle, and λ is the wavelength.

If we assume there is only a single light source, and there is no inter-reflection between the objects in the scene, we can separate the surface reflectance component $L_s(\lambda, i, e, g)$ as the product of a specular power distribution $c_s(\lambda)$ and a scalar $m_s(i, e, g)$. Also we can separate the body reflectance component $L_b(\lambda, i, e, g)$ as the product of a specular power distribution $c_b(\lambda)$ and a scalar $m_b(i, e, g)$. The formula 2.9 can be written as the dichromatic reflection model equation as below:

$$L(\lambda, i, e, g) = m_s(i, e, g)c_s(\lambda) + m_b(i, e, g)c_b(\lambda) \quad (2.10)$$

Thus the light reflected from an object point can be described as a mixture of two distinct spectral power distributions, $c_s(\lambda)$ and $c_b(\lambda)$. Each of these spectral power distributions is scaled according to the geometric reflection property of the body and surface reflection. The body part models conven-

tional matte surfaces, which have Lambertian character. The surface part models highlights, which have the same spectral power distribution as the illuminant. Klinker et al. show that for convex shaped objects with dichromatic reflectance, the distribution of RGBs maps out a T shape, with the bar of the T corresponding to body reflectance, and the stem of the T corresponding to surface reflectance [28]. The dichromatic model has proved useful for a variety of computer vision tasks including colour constancy [32, 31, 20, 50, 57], shape recovery [11] and segmentation [6, 43].

As pointed out by Lee and others, the dichromatic reflectance model proposed by Shafer is inaccurate as a description of certain naturally occurring surfaces [31], and it is not known how well it approximates surfaces in the everyday environment. It is, however, an accurate approximation of a large class of surfaces known as dielectrics, which include paper, paint, and plastics [36].

2.2 Adaptive Colour Modeling

Adaptive colour modeling looks at the problem of how to compensate illumination effect on image appearance in a different way from colour constancy research. Instead of recovering the intrinsic reflectance image from an image taken under unknown illumination, adaptive colour modeling attempts to adjust the colour model according to the illumination change.

Adaptive colour modeling can be divided into two categories, physics-based modeling and statistics-based modeling. In the previous studies of statistics-based modeling, colour is generally represented either by a parametric model or a non-parametric model. A non-parametric colour model is typically a colour histogram [22, 26, 54, 42], and a parametric model is typically a Gaussian model or a Gaussian Mixture model (GMM) in case of multiple colours [63].

59, 64]. The benefit of the colour histogram is that it already represents the underlying probability distribution; therefore there is no need to make any assumption of the underlying probability distribution of the model. The disadvantage is that it normally requires a large number of samples to build such a histogram, which may not always be possible to obtain. Also the evolution of such a histogram due to lighting change is difficult to model. The benefit and the shortcoming of the parametric model are almost contrary to the histogram model, in the sense that it does not require a large number of samples, and its motion is easy to model. Meanwhile, as parametric model is the approximation of the underlying probability distribution, the accuracy of the model depends on the parameters of the model. For example, in the case of GMM, it is important to choose the number of Gaussian components and the attributes for each component.

2.2.1 Non-Parametric Modeling

Non-parametric modeling is a common approach for modeling colour distribution in a colour-based computer vision application. Non-parametric model is appealing because it can be applied to any arbitrary distribution and does not need to make assumptions of the underlying probability distribution function of the model [12].

A commonly used non-parametric approach is based on colour histogram [23, 37, 40, 9]. In a study of tracking multiple people in a real-time video, McKenna used adaptive colour histogram to model the colour of each person in order to track the targets with gradual change in illumination [40]. In this study colour distributions were calculated in the RGB space with $16 \times 16 \times 16$ bins [40], and they were updated based on the previous model and the probability computed at the current time step. Nummiaro presented a study

on tracking object with an adaptive colour-based particle filter [42]. In this study, colour distribution is integrated into a framework of particle filtering. Particle filtering allows to track several hypotheses simultaneously and weight each hypotheses based on a measured similarity to the target model. This allows tracking objects in cases of clutter and occlusions. The target model used in this study is a colour distribution calculated in the RGB space using $8 \times 8 \times 8$ bins, and the target model is updated iteratively using the previous model and the measured model at each image frame in order to adapt to illumination change [42].

Colour histogram is an appropriate model in a problem with large data set and coarsely quantized colour space [40]. Since obtaining large sample data set is not always feasible, and maintaining large sample data set requires large memory storage, kernel density estimation techniques have been introduced to estimate probability distribution without explicitly storing the complete data set [12]. The kernel density estimation estimates the underlying probability function as

$$\hat{f}(x) = \sum_i \alpha_i K(x - x_i) \quad (2.11)$$

where K is a kernel function centered at the data points, usually K is a Gaussian function. $x_i, i = 1 \dots n$ are the sample data points, and α_i are the weighting coefficients at x_i [12].

Kernel density estimation is different from parametric modeling of the probability distribution, in the sense that it does not attempt to fit the data into a parametric model. Although it is a general and effective approach in modeling the underlying probability distribution, it introduces the problem of high computational costs. In the case of M source data points and N target data points, the complexity of evaluating the probability distribution over the N

target data points is $O(MN)$. While this high computational complexity becomes a huge burden for a practical colour tracking or colour segmentation application, Elgammal presented a kernel density estimation approach which uses Fast Gaussian Transform [12] to improve the complexity to $O(M + N)$. This approach is successfully applied in a practical real-time colour segmentation and colour tracking application [12].

Jones carried out a study in which he constructed statistical colour models based on a huge dataset of nearly one billion labeled pixels collected from the World Wide Web [23]. In his study he compared the performance of histogram model and Gaussian mixture model in skin-colour detection, and the results show that histogram model gives slightly better performance than Gaussian mixture model in terms of accuracy. Also, histogram model is superior compared with Gaussian model with respect to the computational costs, though Gaussian model is better with respect to the storage costs [23]. These results are not surprising given the huge amount of training data to construct the colour histogram model.

There is not always a clear distinction between colour histogram model and parametric model. An interesting study tracks human skin colour based on the affine transformation of the skin colour evolution due to dynamic lighting [53]. The colour distribution obtained at two continuous time steps is observed to follow an affine transformation. Once the affine transformation for each time step is calculated, it is fitted into a second order dynamic model. With this dynamic model and the histogram at time step t , the expected histogram at the next step $t + 1$ is then computed. This study is based on the assumption that probability distribution of the skin colour model is a single multivariate normal distribution. Although the colour model is represented with a colour histogram, the evolution of the colour model is described with the means and

the eigenvectors of the distribution.

2.2.2 Parametric Modeling

Gaussian

Gaussian is a commonly used colour model in many colour-based computer vision applications [59, 64, 10, 19]. Wren et al. use a single Gaussian in the YUV colour space to model the colour distribution of a uniformed colour region [59]. Darrell et al. use a single Gaussian to model the human skin colour in the log colour-opponent colour space, which they defined in the form $(\log(G), \log(R) - \log(G), \log(B) - (\log(R) + \log(G))/2)$ [10].

Most of these studies use Gaussian to model human skin colour for face tracking or face detection. According to a study by Yang, under a certain lighting condition, skin-colour distribution can be modeled by a multivariate normal distribution in the normalized colour space [64].

While a single Gaussian has been commonly used to model skin-colour, more studies show that the skin-colour distribution is often multimodal, and can not be adequately represented as a single Gaussian in colour space [55, 65]. In one of his studies on skin-colour distribution, Yang proposed that a Gaussian Mixture model is a more appropriate colour model than a single Gaussian in estimating the skin colour distribution [65]. In his work, Yang carried out statistic tests to examine the validity of Gaussian mixture model. Yang performed Hawkins' statistical test on the normality and homoscedasticity of the estimated Gaussian mixture models, and he used McLachlan's bootstrap method to test the number of components in the mixture model. Yang also presented the results of applying estimated Gaussian mixture density functions in a face detection application. The results show that a mixture model has a better performance than a single Gaussian model in detecting faces.

A single Gaussian model is also used in a general colour tracking application [45]. In this work, a single Gaussian is used as colour model for object with surface nearly Lambertian and of homogeneous colour. This model is derived from the dichromatic reflection model of Klinker [29], which explains the colour distribution of an object with both Lambertian and specular reflectance as a mixture of two clusters, which represent specular component and diffuse component respectively. Assuming specularity is negligible, they found that a single tubal cluster can capture the colour distribution very well. Assuming a simple ellipsoidal colour model, they apply principal component analysis (PCA) to find the bounding ellipsoid of the colour distribution. This simple yet efficient approach is successfully used for tracking coloured objects in real-time video.

From above, we can see that a single Gaussian model is typically applicable to objects with an uniform colour surface, and the object surface property and illumination condition property need to ensure that the specular reflection from the object surface is negligible. This prevents the single Gaussian model from being used in a more general colour-based computer vision application, in which a Gaussian mixture model might be more appropriate. The next section will explain Gaussian mixture model and its applications in computer vision.

Gaussian Mixture Model

As mentioned in the previous section, a Gaussian mixture was presented as a more appropriate model for skin-colour distribution [65]. A Gaussian mixture is also introduced into a general colour-based computer vision, in which the Gaussian mixture is usually used to model a colour region with a mixture of colours [63, 44].

Raja and McKenna presented an adaptive colour tracking algorithm which uses Gaussian mixture model as a parametric colour model [63]. Within this approach, a number of Gaussian functions are used to approximate the multimodal colour distribution, and the conditional probability for each pixel is calculated based on these Gaussian functions.

In Raja's study the purpose of using GMM is to model colour distribution of a multi-coloured object. Suppose we have a multi-coloured object O , and the colour distribution of O can be modeled as a Gaussian mixture model of M Gaussian components. Given a pixel ξ , the conditional density for ξ to belong to O is

$$p(\xi|O) = \sum_{j=1}^M p(\xi|j)P(j) \quad (2.12)$$

Where $P(j)$ is the prior probability that pixel ξ is generated by component j , and $\sum_{j=1}^M P(j) = 1$. If we assume that each Gaussian component has a mean μ_j and a covariance matrix Σ_j ,

$$p(\xi|j) = \frac{1}{2\pi|\Sigma_j|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(\xi-\mu_j)^T \Sigma_j^{-1}(\xi-\mu_j)} \quad (2.13)$$

Raja uses a standard Expectation-Maximization (EM) algorithm to fit the GMM to a data set. The focus of his work is on the model order selection problem, which is how to choose the number of parameters of the GMM in order to have an accurate modeling of the underlying distribution. Raja presents an algorithm which chooses the number of parameters by iteratively splitting components and evaluating with a validation set. It starts with modeling with a GMM of low order, then iteratively finds the component with the lowest responsibility and splits this component into two components. After each splitting, the likelihood for the validation set is evaluated. The iteration ter-

minates when the peak of the likelihood measurement for the validation set is detected.

The adaptation of the colour model is done by linear extrapolating the GMM obtained at time step $t - 1$ and the measured GMM at time step t from the labeled pixels. The resulting GMM is used as the updated colour model for time step t , which is expected to adapt to the changes due to dynamic lighting.

Raja's work has been extended by many other researchers. In a related study, instead of making implicit assumption about the linear transformation of the GMM, they use transductive learning techniques to train the classifiers with the labeled pixels obtained during the tracking [60]. A recent study presents an adaptive skin-detection method [66]. Given an image, they derive GMM from a coarse skin colour classification and then use a SVM classifier to identify the skin Gaussian from the GMM by incorporating spatial and shape information of the skin pixels.

Raja treats the specular highlights as outliers points during the modeling process, and he claimed that these outliers points are possibly caused by the image noise and specular highlights have little influence upon the mixture model [63]. While in our work, specular highlights are innegligible component of the colour distribution, and our GMM takes into account the specular component of the colour distribution.

2.3 Colour Vision in RoboCup

2.3.1 Summary

The Robot World Cup Initiative (RoboCup) is an international research initiative that uses the domain of robot soccer to foster AI and intelligent ro-

botics research [25]. RoboCup serves as an excellent test bed in several areas of research interests, such as path planning, obstacle avoidance, multi-agent collaboration, game strategy, real-time data and image processing, robotic vision, artificial intelligence and control [27]. In recent years, RoboCup has gained much popularity among researchers worldwide. It is the colour vision in RoboCup that motivates the work on adaptive colour classification.

In the RoboCup small-size league, two teams, each made up of five robots, play soccer with an orange golf ball on a field 3.4 meters wide by 4.9 meters long. Robots must fit inside a 180 mm diameter cylinder, with a maximum height of 150 mm if the team is using the global vision system. Each team is allowed to mount one or more cameras above the field to provide a global vision system [49]. The global vision system is the only sensor for the team, through which it sees the entire field and detects the positions of the team members, the opponent players, and the ball. The information is fed to the next module which decides the game strategy, and sends radio signals to each player to give them instructions with respect to how to play the game.

As the vision system acts as the “eye” for the entire team, it is important that the vision system detects the positions of the ball and the players from both teams accurately, otherwise game strategy would be misled if incomplete or even false information is provided by vision system. Also the vision system needs to be efficient to handle 30 frames/sec NTSC video signal so that the process of making decisions on game strategy does not suffer from the time of processing with vision. Colour vision is used as it can serve the requirement of both efficiency and accuracy very well. According to the game regulation, each of the two teams is assigned a colour before the game, namely yellow or blue. Circular markers of the assigned colour must be mounted on top of the robots, each with a diameter of 50 mm [49]. The only other colours permitted

are pink, green, light cyan, black, and white. Most of the teams use a certain pattern of colour patches in these colours on the top of each of the robots to indicate the identity and the heading of the robot. For Team Canuck of the University of Alberta, each robot has a pattern of three colour patches on the top surrounding the center colour patch which is of the team colour. Two of the patches are collinear with the center patch and parallel with the robot's heading, and the third is perpendicular to the robot heading and on the right hand side of the center patch, as viewed from behind the robot. Each patch is either green or pink, representing "1" or "0" in binary coding. The heading of the robot can be computed based on the configuration of the three colour patches. With the heading identified, the identification number of the robot is the binary number represented by the colour and order of these three patches.

With these colour patch configuration, the goal of the vision system is to identify a number of colour blobs in several different colours from each image frame of the real-time video sequence. Once the blobs are all identified, the vision system will be able to locate the ball and each of the robots on the field. Therefore, the task of the vision system is colour classification of multiple colours in real-time video.

2.3.2 Vision system of previous teams

The complexity of the colour classification problem with RoboCup was historically rather straight-forward, as the field used to be as small as 2.3m wide by 2.8m long, under fairly well controlled uniform room lighting. In such a setting, the colour model for each colour normally does not suffer from the influence of temporal illumination change. A carefully calibrated colour model for each colour is sufficiently accurate during the match. The spatial variation of illumination is quiet small and can be easily dealt with. For example, one

approach to compensate spatial illumination variation is to divide the entire field into a 4×4 grid and assign each region a scalar multiplication factor on the R, G, and B three channels of the pixel within this region [41].

As the complexity of the colour classification problem is not a major concern, most of the efforts put on the vision system are focused on the efficiency issue. A commonly used approach for modeling the colour class is using a lookup table (LUT).

In the commonly used RGB colour space, a pixel has three colour components, each has a discrete value of $[0..255]$. The pixel value takes the form of $RGB = [0..255]^3$, a total of 2^{24} distinct pixel values compose the entire colour space. With respect to how to define a colour class, a colour class can be regarded as a subset of the entire colour space, a set of all possible pixel values of the same colour. A LUT indexed by colour component values is commonly used to model the colour class. The LUT can be pre-calibrated by collecting sample pixels of the same colour and then populating the LUT entries with these sample pixel values. Once a LUT is built, we can answer a query that “does a given pixel p_i belong to colour class C_k ?” by looking up this pixel in the LUT using the pixel values as indexes. This operation of checking pixel membership is very efficient as it only spends CPU time on a few machine instructions. While it is very efficient, the LUT method has the inherited problem of colour space quantization. In RGB colour space the colour component of each channel has 256 discrete values, so the number of all the entries in the LUT can be as large as $256 \times 256 \times 256$. The large size of LUT is not desirable because the LUT needs to be stored in memory in order to ensure efficient pixel membership checking. The huge memory burden can reduce the efficiency of the system.

Many different approaches have been taken to deal with the quantization

problem of LUT. One common approach is to reduce the dimensionality of the colour class in order to reduce the storage space. A simple approach defines a rectangular bounding box around the colour class in 3D colour space, and it classifies a pixel according to if this pixel is inside the box or outside [21]. This bounding box is also called axis-aligned rectangular volumes. Within this approach, a total of 6 numbers is sufficient to define the membership of a pixel:

$$p_i \in C_k \equiv (R_{p_i} \in [r_{lo}, r_{hi}]) \& (G_{p_i} \in [g_{lo}, g_{hi}]) \& (B_{p_i} \in [b_{lo}, b_{hi}]). \quad (2.14)$$

The essence of this approach is to reduce the dimensionality of the colour space from 3D to 1D. Using two numbers on each of the three dimensions to represent the colour class is equivalent to projecting the 3D colour class onto three axes to get a 1D representation. As simple as it looks, this approach has the problem of over-generalization. In RGB colour space, a colour class which consists of a set of pixels of the same hue but different brightness does not align with axes well. Therefore the rectangular volume is inappropriate to specify this colour class [41]. Imagine an extreme case in which we want to model white colour, as the illumination brightness increases from totally dark to very bright, the pixels values of white colour evolves along the diagonal axis of the RGB colour cube from $[0, 0, 0]$ to $[255, 255, 255]$. Now the axis-aligned rectangular volume is as big as the entire colour cube, all possible colours inside the colour cube will be classified as white.

It is observed that in some colour spaces a colour class has better alignment with the axes. YUV space and HSI space are two commonly used colour spaces which have this property. In YUV space, Y component indicates brightness or luminance, while U and V components indicate colour or chrominance. In HSI

space, I component indicates brightness or luminance, while H and S components indicate colour or chrominance. As the illumination brightness changes, the colour class evolves mostly along the luminance axis, while the variabilities along two chrominance axes are comparatively small. The colour class of pixels with the same hue but different brightness aligns with luminance axis approximately, making the rectangular volume representation more accurate. Most colour cameras provide video input in RGB space. Because the conversion from RGB to HSI is computationally expensive and the conversion from RGB to YUV is much efficient, YUV is a more appealing candidate colour space in real-time vision applications. Carnegie-Mellon's RoboDragons team uses axis-aligned rectangular volumes in YUV space for colour classification [8].

Although choosing an appropriate colour space improves colour classification, as pointed by Thomas et al. The rectangular representation of the colour model is quite restrictive, however, since any incident luminosity changes and glare from incorrect field lighting produce colour deviations beyond rectangular decision regions [56]. A colour class representation more accurate than rectangular volume is necessary to handle these situations.

Other approaches either use LUT in a coarsely quantized 3D colour space, or use LUT in a colour space with a reduced dimensionality. The vision system used by CM-Dragons'01, CMVision, segments pixels in YUV colour space using a 3D lookup table, which is a subsampled version of the colour space with 8 levels for Y, and 32 levels each for U and V [7]. This coarsely quantized LUT does not require huge memory size. Meanwhile it keeps a fairly accurate representation of the colour class. The Dutch team uses two lookup tables, one for Y and another for (U,V), such that a pixel is classified as in a colour class if it exists in both lookup tables [24]. This is equivalent to reduce the

class is discarded after projecting it on 2D planes, the colour class is approximated with the 3D shape reconstructed from three 2D projections. While this approach has been successfully used for accurate and fast colour classification, it should be noted that since this approach uses reconstructed 3D shape to approximate a colour model, the accuracy of the colour model depends on how well the colour class aligns with the axes of colour space. This reconstructed colour class is a better approximation when it aligns well than the case when it does not align well. This suggests that a different colour space such as YUV or HSI could improve the accuracy of the colour model of this approach.

While most of the studies on colour vision in RoboCup only deal with static colour classification, there is some work which aims at adaptive colour classification in order to be able to adapt to varying light [17, 34, 33].

Wyeth presented an adaptive vision for robot soccer [17]. In their approach, first the background playing field is removed so that the points of interest are highlighted. This background removal is based on the fact that the playing field is of a uniform colour and it occupies a major portion of the image. Therefore if we take the colour histogram of a field image, the three peaks in three component colours represent the colour histogram of the field colour. If we define a cube around the three peaks, we can classify pixels within the cube as field, and pixels outside the cube as points of interest. This technique is called adaptive colour histogram, in the sense that even though the lighting change causes the colour histograms to shift, the peaks always remain to represent the field colour. Therefore, if we can derive the cube around the field colour for each image frame, we can differentiate field from points of interest even under lighting change. This step is followed by a two pass thinning operation to reduce the noise, and then a two pass dilation operation to fill the holes on the robots if the marker on a robot was discarded because it was similar

to field colour. The selected pixels are aggregated into blobs, followed by a template matching to examine if a blob is the ball or the colour marker for a team robot or an opponent robot. The final step is to get orientation and member identification of the team robots.

Li presented an adaptive colour segmentation algorithm for local vision of Sony legged robots [34]. The camera in the Sony legged robot outputs images in a YUV format and it has a hardware colour segmentation method, called a General Colour Detection (GCD) Table. GCD conditions can be described as

$$\begin{aligned}
 i &= P_Y/8 \\
 T_{U \min i} &< P_U < T_{U \max i} \\
 T_{V \min i} &< P_V < T_{V \max i}
 \end{aligned} \tag{2.17}$$

There is a rectangular threshold in 2D UV space, which is indexed by the value of Y component. Using a learning network trained with manually constructed GCD tables under a set of lighting conditions, the algorithm can generate GCD tables under varying illuminations. The learning network has Y, U, V, L1, and L2 as inputs, L1 is the luminance decided by the meter, and L2 is the luminance measured from the image. The output is the CDT tables when illumination changes [34].

Chapter 3

Methodology

In this thesis, we study the problem of adaptive colour classification. This study is motivated by the robocup research initiative. The colour vision of the robocup is essentially a colour classification system, as the major functionality of the robocup vision, target recognition, is based on the result of colour classification. In this problem domain, colour classification is done on each image frame of a real-time video sequence, such that a number of blobs with several different colours can be located in the image. These colour blobs are used as markers of the robots and the ball, and the locations of these colour blobs serve as cues to help locate and identify each robot player and the ball during the game. Figure 3.1 shows a typical view of the field with the robots and the ball from the camera of the vision system. For team Canuck, each robot has four colour patches on the top. The center patch is either blue or yellow, which represents the team colour. The other three patches are in either pink or green, which is the binary coding representing the player number.

Since there is spatial variation of the illumination on the field which affects the colour appearance of each blob at different locations on the field, this spatial variation of the illumination needs to be taken into account in order to locate multiple targets scattered over the field at each time step. Also, there is possible temporal variation of the illumination on the field which affects the

colour appearance of each blob as well. Therefore, both spatial variation and temporal variation of the illumination need to be considered for an accurate colour classification, and they both are dealt with in our algorithm.

Our work follows adaptive colour modeling approaches. As mentioned above, there are two major issues existing in this colour classification problem, the spatial illumination variation and the temporal illumination variation. Our algorithm deals with these two issues in this way: in order to deal with spatial illumination variation, at each time step we construct a global colour classifier which is capable of classifying the blobs located at different positions on the field; in order to deal with the temporal illumination variation, we update the global colour classifier according to the illumination change so that the classifier can adapt to dynamic illumination and accurately classify the blobs in the consequent image frame.

There are several issues with respect to an adaptive colour modeling approach. The first issue is how we model the colour. Based on the property of our problem domain, we decide to use a parametric model to represent the colour distribution. The second issue is which colour space we choose. Based on our comparison studies and the previous work from other researchers, we choose YUV colour space in our algorithm. The third issue is how to update colour model as illumination changes temporally. We choose an exponentially decaying function as our adaptation mechanism.

This chapter describes the methodology of our algorithm in detail, with respect to the issues mentioned above.

3.1 Colour Modeling

Under an invariant illumination, an accurate colour classification requires a colour model which can represent the colour distribution completely and ac-

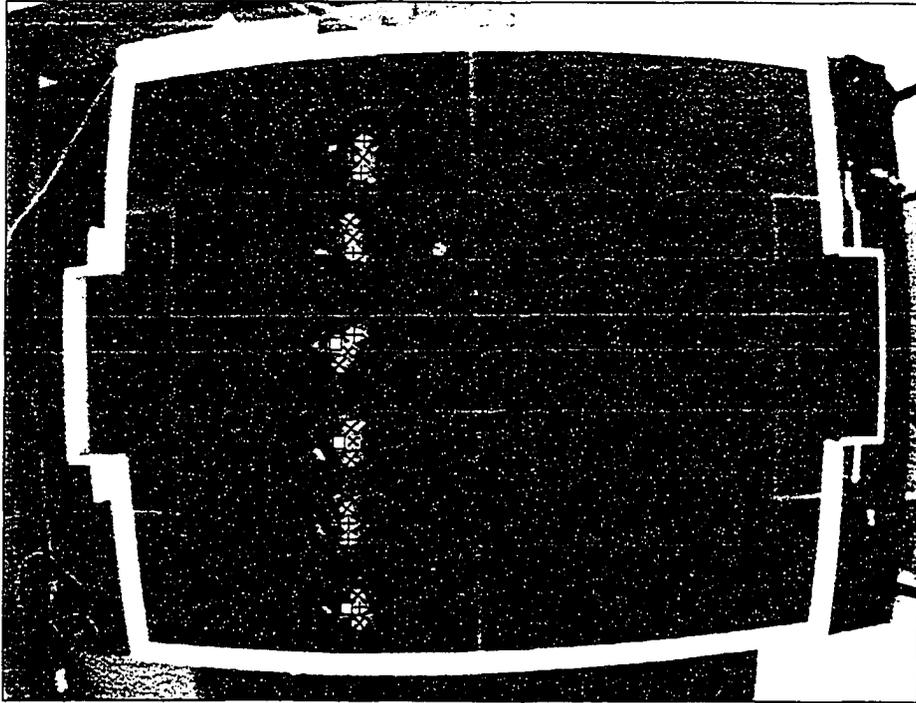


Figure 3.1: A typical camera view of robocup team and the field. Pink, green, blue, and yellow blobs are illustrated with \times , $+$, circle, and square.

curately. An over general colour model will likely lead to an over-classification with a large number of false positives. The falsely classified colour pixels could reduce the performance of the system. On the other hand, an incomplete colour model will likely cause an under-classification with a large number of false negatives, and possibly not be able to classify the interesting regions in the image. In our case, we need to avoid false negatives and false positives as much as possible, because the accuracy of the vision system is essential. Either not being able to locate a robot or the ball or misjudging the background noise colour as a robot or the ball will very likely lead to making mistakes in game strategy, and it will jeopardize the entire game.

We choose parametric modeling instead of non-parametric modeling for each colour. A non-parametric modeling normally requires a large number of sample points to construct the model, while in our case it is not feasible to

obtain a large number of samples, since each colour blob is of very small size in terms of the number of pixels it occupies in the entire image. Take the orange ball as an example, the number of pixels extracted from the ball from an image frame usually does not exceed 40 or 50. While this small number of points makes non-parametric modeling unsuitable, parametric modeling is more suitable in our case since it does not require large number of samples. Another advantage of parametric modeling is that each of the colour blobs to be classified consists of a uniform colour, and the uniform colour appearance greatly reduces the complexity of the parametric model needed to represent the colour distribution of the object.

Now the question is which parametric model we should use in order to model the colour distribution in an accurate and complete manner. For a uniform colour such as human skin-colour or an object of homogeneous colour, most of the previous works assume that the underlying probability distribution of the colour model is a single Gaussian. This is based on the assumption that the object surface is approximately Lambertian. Based on this assumption, a single Gaussian can be used as a sufficiently accurate colour model. Also the evolution of the colour model due to dynamic lighting can be approximated as a linear transformation. While the assumption of Lambertian surface is reasonably accurate in many applications, they do not hold for a general purpose colour modeling. Most objects existing in the real world do not have an ideally Lambertian surface; instead it is very common that they have a surface of inhomogeneous materials, such as paints, plastics, and paper. For these objects, the colour distribution of their surfaces can not be modeled as a single Gaussian. The dichromatic reflectance model explains the colour distribution of these materials as a “T-shape” of two clusters [28, 51, 50]. In our case, it is observed that the colour distributions of the colour patches on robots are

not suitable to be modeled with a single Gaussian; instead they consist of two clusters which are approximately Gaussian-shaped. The colour distributions of the green colour and the pink colour in RGB space under a normal room lighting are given in Figure 3.2, the top figure is pink, the bottom figure is green. These colour distributions are in fact consistent with the dichromatic reflectance model proposed by Shafer and others [51, 50].

According to the dichromatic reflectance model for inhomogeneous dielectrics proposed by Shafer and others [51, 50], the light reflected from a surface comprises two physically different types of reflection, surface reflection and body reflection. The body part models conventional matte surfaces, which has Lambertian property. The surface part models highlight, which has the same spectral power distribution as the illuminant. Klinker et al. show that for convex shaped objects with dichromatic reflectance, the distribution of RGBs maps out a T shape, with the bar of the T corresponding to body reflectance, and the stem of the T corresponding to surface reflectance [28]. The colour histogram is shown in Figure 3.3.

Figure 3.4 shows an example of the colour distribution of the green patches in RGB space under different lighting conditions. A number of robots with green patches on their top scatter over the field, and they are exposed under four different lighting conditions as the room lighting is dimmed from bright to dark. Under each lighting condition, the pixels are manually selected from the green blobs in an image frame. The selected pixel values are fitted into a two-component Gaussian mixture model, and the pixels belonging to each of the Gaussian components are labeled as red or green in this figure. From the left to the right, the lighting condition is gradually changed by dimming the room lighting from bright to dark. It is observed that data points can be fitted into a mixture of two Gaussian components well. This is especially the case when

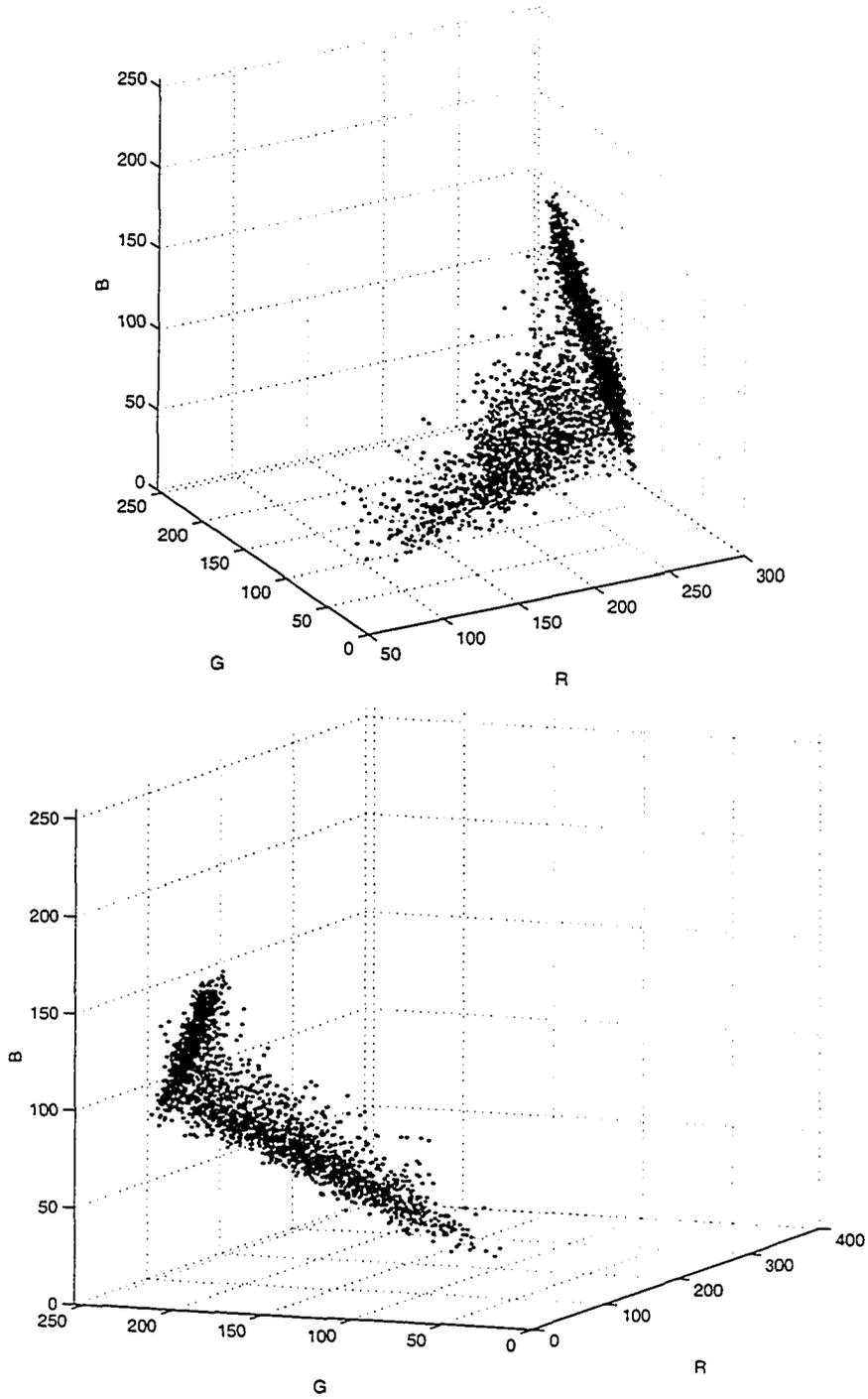


Figure 3.2: Colour distribution of the colour patches used on top of the robot in RGB space

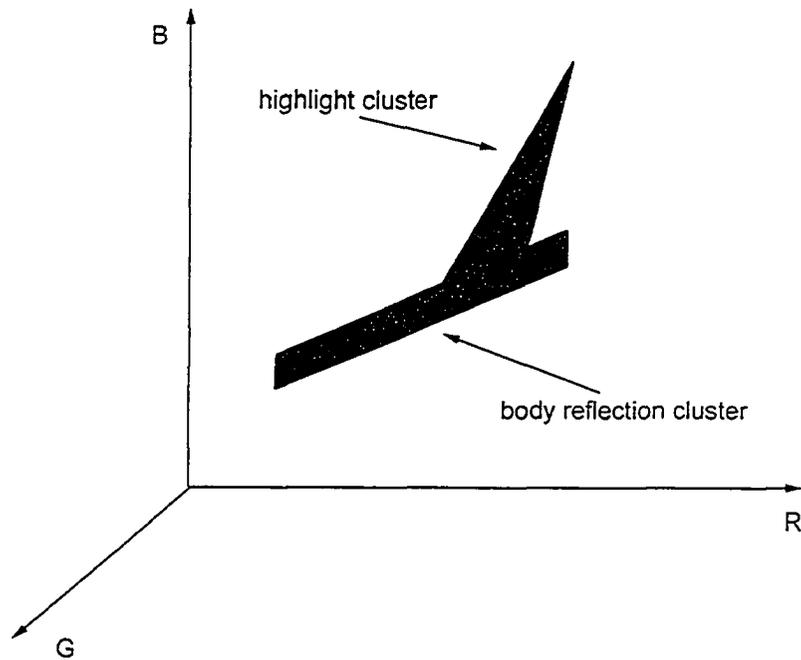


Figure 3.3: Histogram of an object with dichromatic reflectance

the illumination is bright, as shown in the top two figures. As illumination intensity decreases, the distinction between these two Gaussians gradually decreases, as one of the Gaussians starts to grow smaller, as shown in the bottom two figures, but even so there still clearly exist two Gaussian components. The colour distribution shown in this figure is consistent with Klinker's T shape model, the stem of the T corresponding to surface reflectance, and the bar of the T corresponding to body reflectance. Based on our observation, we expect that a two-component Gaussian mixture models the colour distribution accurately.

Another observation is, throughout this process as the room lighting is dimmed from bright to dark, the portion of the surface reflection decreases and the portion of the body reflection increases. Also the colour distributions of both the body reflectance and the surface reflectance go through a wide

range in colour space under dynamic lighting. This demonstrates that a single Gaussian model is inadequate to represent the colour distribution, also it is unsuitable to model the evolution of the colour distribution due to dynamic lighting. As both the surface reflection and the body reflection should be taken into account in order to have an accurate colour classifier under a stable lighting condition, adaptation needs to be done for both of surface reflection and body reflection in order to adapt to dynamic lighting changes.

For such a colour distribution, either representing it with a single Gaussian or modeling its evolution with a linear transformation is no longer suitable. The fact that this colour distribution has dichromatic reflectance property and it is a “T-shape” of two clusters motivates us to model the colour distribution with two Gaussian components, each component corresponding to one of the two clusters.

In our algorithm, we choose GMM to represent the colour distribution based on the dichromatic reflectance model. The dichromatic reflectance model explains a colour distribution as the combination of two clusters: diffuse cluster from the body reflectance and specular cluster from the surface reflectance. We use GMM of two components to represent these two clusters. Again the benefit of using GMM is that it does not require many samples, which is particularly important in our case, since the colour blobs on the field are of very small size.

We use a standard Expectation-Maximization (EM) algorithm to derive the GMM from the set of sample pixels. Given a set of sample pixels $\{y_i, i = 1 \dots n\}$, a standard EM is a commonly used approach to provide an effective maximum-likelihood method to fit a GMM to the data set [2, 46]. The EM used in our algorithm is based on the Cluster algorithm, an unsupervised algorithm for modeling Gaussian mixtures, presented by Bouman at Purdue University [3].

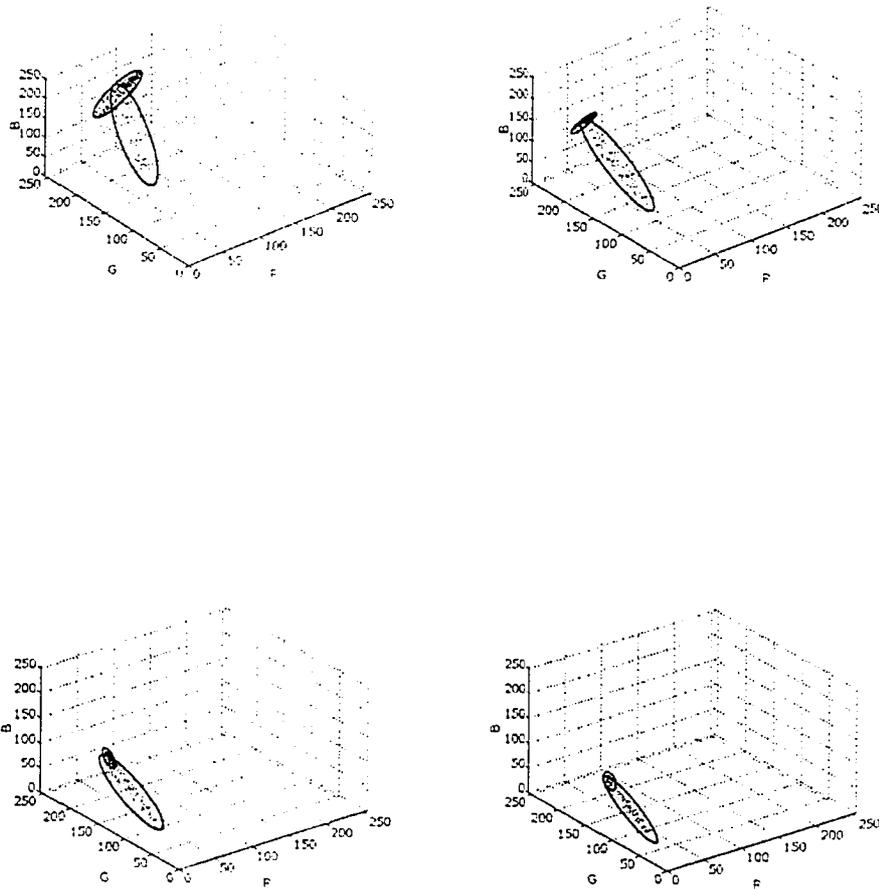


Figure 3.4: The evolution of the distribution of green colour in RGB space

Instead of using maximum-likelihood estimation, the EM algorithm in our algorithm uses the Rissanen order identification criterion known as minimum description length (MDL) to estimate the number of clusters[3]. The benefit of this EM algorithm is that it is equivalent to maximum-likelihood (ML) estimation when the number of clusters is fixed, but in addition it allows the

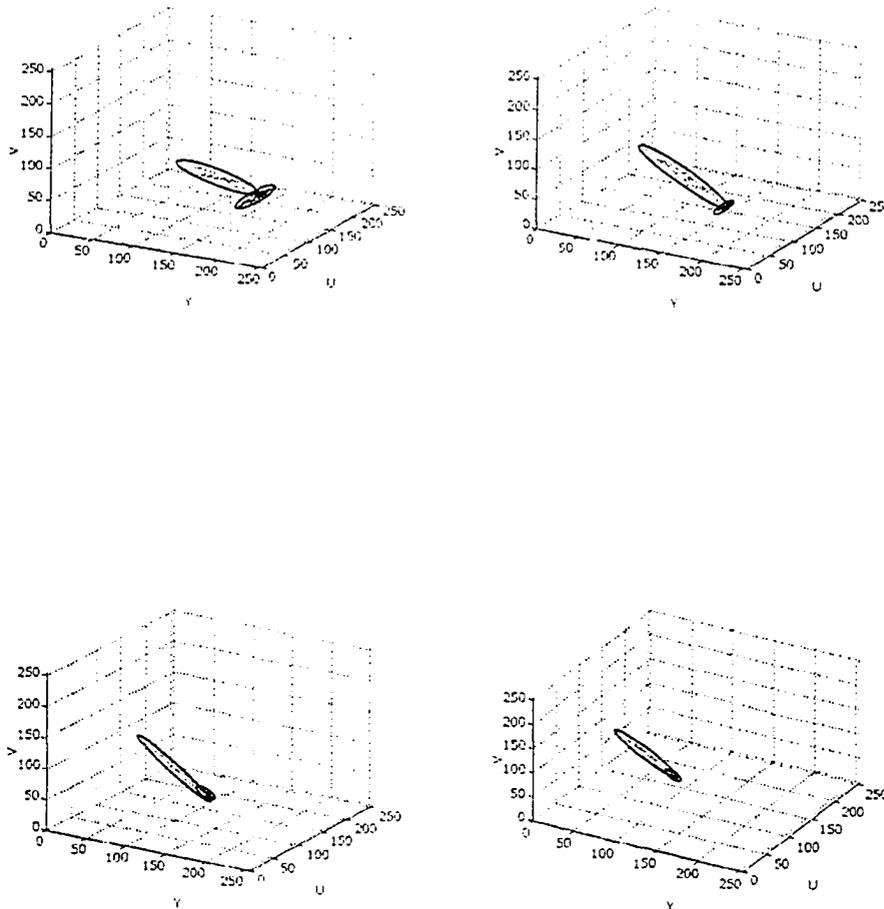


Figure 3.5: The evolution of the distribution of green colour in YUV space number of clusters to be accurately estimated. This is important in our case because the ability of an accurate modeling order estimation ensures the robustness of our algorithm. Although through our experiments it is observed that a GMM of two Gaussian components is an accurate model of the colour distribution in most cases, it could also happen that a single Gaussian repre-

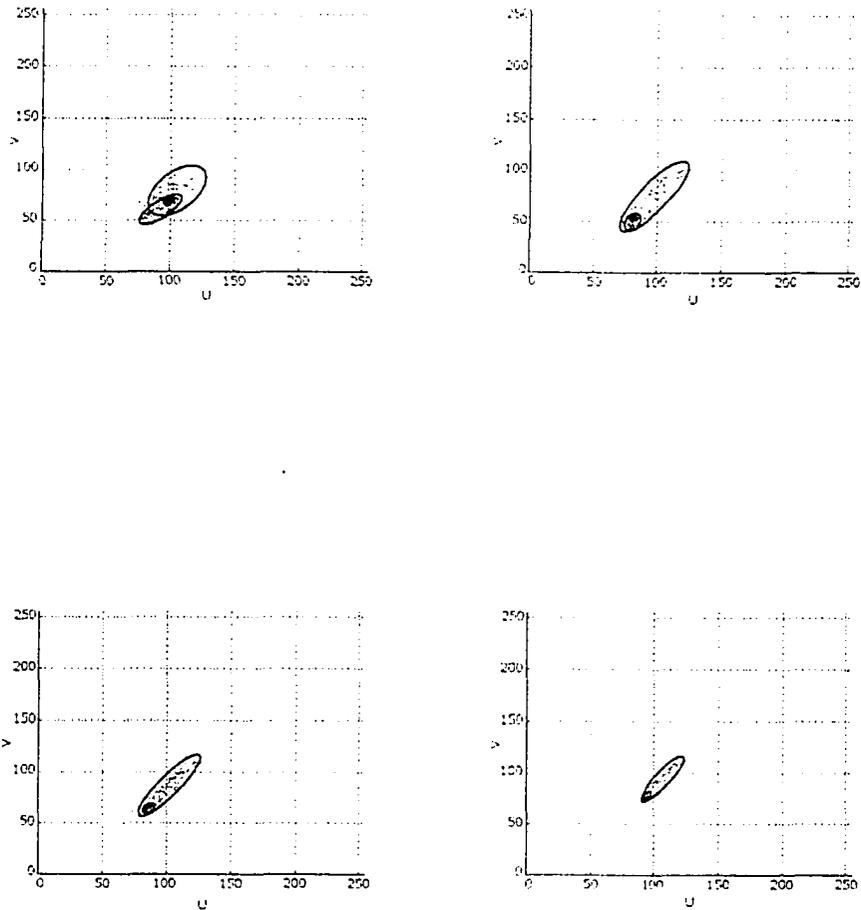


Figure 3.6: The evolution of the distribution of green colour in UV space
 representation of the colour distribution is appropriate under certain illumination conditions. For example, when the illumination intensity drops to a certain level, the surface reflectance of all the surface patches diminishes to an insignificant portion of the entire colour distribution. Now the dominant part of the colour distribution is the body reflectance, which can be modeled with

a single Gaussian. In this case, forcing the modeling to be two components could possibly introduce the problem of over-clustering and therefore increase the error in modeling. Being able to estimate the accurate order for modeling ensures the robustness and the stability of our algorithm under a broad range of illumination conditions.

EM is an iterative optimization method to estimate some unknown parameters Θ , given measurement data U , where the measurement data are incomplete or the likelihood function involves latent variables or so-called hidden variables. Intuitively, what EM does is to iteratively grow the data by guessing the values of the hidden variables and to iteratively estimate the parameters by assuming that the guessed values are the true values. Informally, the EM algorithm starts with randomly assigning values to all the parameters to be estimated. It then iteratively alternates between two steps, which are called the expectation step and the maximization step, i.e., the E-step and the M-step respectively. In the E-step, given the current settings of parameters and our observed incomplete data, it computes the expected likelihood for the complete data (the Q-function) where the expectation is taken with respect to the computed conditional distribution of the latent variables. In the M-step, it re-estimates all the parameters by maximizing the Q-function. Once we have a new generation of parameter values, we can repeat another iteration of the E-step and the M-step. This process continues until the likelihood converges, i.e., reaching a local maximum.

In summary, the general procedure of the EM algorithm is the following:

1. Initialize $\theta^{(0)}$ randomly or heuristically according to any prior knowledge about where the optimal parameter value might be.
2. Iteratively improve the estimate of θ by alternating between the following

two-steps:

- (a) The E-step (expectation): Compute $Q(\theta; \theta^{(n)})$
- (b) The M-step (maximization): Re-estimate θ by maximizing the Q-function: $\theta^{(n+1)} = \text{argmax} Q(\theta; \theta^{(n)})$

3. Stop when the likelihood $L(\theta)$ converges.

In our method, the colour model is a GMM which consists of at most two Gaussian components. Assume the sample points are M dimensional vectors, the number of Gaussian components is K , and each Gaussian component is the subclass of the colour class, the following parameters are required to completely specify the k^{th} subclass, where $k \leq K, K = \{1, 2\}$:

π_k – the probability that a pixel belongs to k^{th} subclass.

μ_k – the M dimensional spectral mean vector for k^{th} subclass.

R_k – the $M \times M$ spectral covariance matrix for k^{th} subclass.

Also we need to estimate K , therefore the set of parameters of the GMM is $\{K, \theta\}$, where $\theta = \{\pi_k, \mu_k, R_k\}_{k=1}^K$.

First, let us consider the problem of optimizing θ when K is fixed. Let y_1, y_2, \dots, y_N be N pixels of the sample data set. Assume that for each pixel y_i the subclass of that pixel is x_n . Based on the assumption that each subclass has a multivariate Gaussian distribution, the probability density function for the pixel y_n given that $x_n = k$ is given by

$$p(y_n|k, \theta) = \frac{1}{(2\pi)^{M/2}} |R_k|^{-1/2} \exp \left\{ -\frac{1}{2} (y_n - \mu_k)^t R_k^{-1} (y_n - \mu_k) \right\} \quad (3.1)$$

The density function of y_n given the parameter θ is computed by applying the definition of conditional probability and summing $p(y_n|k, \theta)$ over k .

$$p(y_n|\theta) = \sum_{k=1}^K p(y_n|k, \theta)\pi_k \quad (3.2)$$

The probability of the entire sequence $Y = \{y_n\}_{n=1}^N$ is then given by

$$p(Y|K, \theta) = \prod_{n=1}^N p(y_n|\theta) = \prod_{n=1}^N \left(\sum_{k=1}^K p(y_n|k, \theta)\pi_k \right) \quad (3.3)$$

If we take the logarithm of the above equation, it is then

$$\log p(Y|K, \theta) = \sum_{n=1}^N \log \left(\sum_{k=1}^K p(y_n|k, \theta)\pi_k \right) \quad (3.4)$$

The objective is to estimate the parameters K and θ such that $p(Y|K, \theta)$ has the maximum value. The commonly used maximum likelihood (ML) estimate is given by

$$\hat{\theta}_{ML} = \arg \max_{\theta} \log p(Y|K, \theta) \quad (3.5)$$

However, the above maximum-likelihood estimate has a problem when the value of K is not fixed. As pointed out by Bouman, the log likelihood may always be increased by adding more subclasses; i.e., since the likelihood may always be made better by choosing a large number of subclusters, the ML estimate of K is not well defined [3]. In order to estimate the order of the Gaussian mixture, the minimum description length (MDL) suggested by Rissanen is used instead of the maximum likelihood estimate [48, 3]. This descriptor works by attempting to find the model order which minimizes the number of bits that would be required to code both the data samples y_n and the parameter vector θ [3]. An approximate expression developed by Rissanen is

$$MDL(K, \theta) = -\log p(Y|K, \theta) + \frac{1}{2}L \log(NM) \quad (3.6)$$

in which N is the size of the sample data, M is the number of the dimension of the data, and L is the number of continuously valued real numbers required to specify the parameter θ [3]. Now the objective is to minimize the MDL criterion given by

$$MDL(K, \theta) = - \sum_{n=1}^N \log \left(\sum_{k=1}^K p(y_n|k, \theta) \pi_k \right) + \frac{1}{2} L \log(NM) \quad (3.7)$$

The expectation-maximization (EM) algorithm is introduced to minimize the MDL criterion. Intuitively, the EM algorithm works by first classifying the pixels y_n according to their subclasses, and then re-estimating the subclass parameters based on this approximate classification. The process is started by assuming that the true parameter is given by $\theta^{(i)}$, where i is the index of the the iterative procedure of the EM algorithm for improving the MDL criterion [3]. Given $\theta^{(i)}$, the probability that pixel y_n belongs to subclass k may then be computed using the Bayes rule.

$$p(k|y_n, \theta^{(i)}) = \frac{p(y_n|k, \theta^{(i)}) \pi_k}{\sum_{l=1}^K p(y_n|l, \theta^{(i)}) \pi_l} \quad (3.8)$$

Based on these estimated subclass memberships we will then compute new set of estimated parameters for each subclass. If we denote these new estimated parameters as $\bar{\pi}_k$, $\bar{\mu}_k$ and \bar{R}_k , they are given by

$$\bar{N}_k = \sum_{n=1}^N p(k|y_n, \theta^{(i)}) \quad (3.9)$$

$$\bar{\pi}_k = \frac{\bar{N}_k}{N} \quad (3.10)$$

$$\bar{\mu}_k = \frac{1}{\bar{N}_k} \sum_{n=1}^N y_n p(k|y_n, \theta^{(i)}) \quad (3.11)$$

$$\bar{R}_k = \frac{1}{\bar{N}_k} \sum_{n=1}^N (y_n - \bar{\mu}_k)(y_n - \bar{\mu}_k)^t p(k|y_n, \theta^{(i)}) \quad (3.12)$$

The EM algorithm update equation is computed as the following [3]:

$$Q(\theta; \theta^{(i)}) = E[\log p(y, X|\theta)|Y = y, \theta^{(i)}] - \frac{1}{2}L \log(NM) \quad (3.13)$$

where Y and X are the sets of random variables $\{y_n\}_{n=1}^N$ and $\{x_n\}_{n=1}^N$ respectively, and y and x are realizations of these random objects [3].

It is proved by Baum and etc. that for all θ in EM algorithm [1],

$$MDL(K, \theta) - MDL(K, \theta^{(i)}) < Q(\theta^{(i)}; \theta^{(i)}) - Q(\theta; \theta^{(i)}) \quad (3.14)$$

This shows that any value of θ that increases the value of $Q(\theta; \theta^{(i)})$ is guaranteed to reduce the MDL criteria. Therefore optimization of MDL can be replaced by optimization of $Q(\theta; \theta^{(i)})$ [3]. The objective of the EM algorithm is to find a local minimum of the MDL function by iteratively optimizing with respect to θ [3]. We can accomplish this goal by finding the local maximum of $Q(\theta; \theta^{(i)})$.

A more explicit form for the function $Q(\theta; \theta^{(i)})$ after substituting $\log p(y, X|\theta)$ and simplifying is provided by Bouman as the following[3]

$$\begin{aligned} Q(\theta; \theta^{(i)}) = & \\ & \sum_{k=1}^K \bar{N}_k \left\{ -\frac{1}{2} \text{trace}[\bar{R}_k R_k^{-1}] - \frac{1}{2} (\bar{\mu}_k - \mu_k)^t R_k^{-1} (\bar{\mu}_k - \mu_k) \right. \\ & \left. - \frac{M}{2} \log(2\pi) - \frac{1}{2} \log(|R_k|) + \log(\pi_k) \right\} - \frac{1}{2} L \log(NM) \end{aligned}$$

The optimization of θ is done using the following update equations [3]

$$\begin{aligned} & (\bar{\pi}^{(i+1)}, \bar{\mu}^{(i+1)}, \bar{R}^{(i+1)}) \\ & = \arg \max_{(\bar{\pi}, \bar{\mu}, \bar{R})} Q(\theta; \theta^{(i)}) \\ & = (\bar{\pi}, \bar{\mu}, \bar{R}) \end{aligned}$$

Above we have shown how to update the parameter θ for a given K , now the question is how to find the appropriate K as the model order. In Cluster an agglomerative clustering technique is used to find the accurate K . Given an

initial K , the agglomerative technique is to combine some of the subclusters together to form a clustering with less clusters. Finding K is a much simplified problem in our case, as $K \leq 2$. As mentioned before, this initial value of K is based on the empirical knowledge of the problem domain. First of all, EM is executed with the initial $K = 2$ to find the optimized parameters, and MDL for $k = 2$ is computed. It then follows that $K = 1$ is applied to clustering to find the optimized θ , and MDL for $k = 1$ is computed. The appropriate K which minimizes the value of MDL is then chosen, together with the corresponding parameter of θ .

3.2 Colour Space

Our system uses YUV colour space for colour classification. YUV space is commonly used in a computer vision system mostly due to the fact that it is capable of dealing with mild lighting variation. Also, YUV space is especially popular in the robocup colour vision domain.

YUV and YIQ are standard colour spaces used for analogue television transmission. YUV is used in European TVs (PAL) and YIQ in North American TVs (NTSC). Y is the luminance component and is usually referred to as the luma component, which comes from CIE standard. U,V or I,Q are the chrominance components, which are the colour signals.

The benefit of YUV space is that Y channel, which represents luminance, captures most of the variations due to the luminance change of the environment, while U and V channels which represent chromaticity are insensitive to the luminance change. This property of the YUV space makes it easier to model the evolution of the colour distribution due to illumination changes. Another benefit of YUV space is that conversion from RGB camera input to YUV space is relatively inexpensive, compared with other space such as HSI.

Some cameras even have YUV input, which minimizes the time of colour space conversion.

$$\begin{aligned}
 Y &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\
 U &= -0.147 \times R - 0.289 \times G + 0.436 \times B \\
 V &= 0.615 \times R - 0.515 \times G - 0.100 \times B
 \end{aligned}
 \tag{3.15}$$

Our observation shows that YUV is a suitable colour space for our problem. Figure 3.5 and Figure 3.6 show the colour distribution of green patches under different lighting conditions. These two figures are under the same experiment conditions as Figure 3.4, except in YUV and UV space respectively. From Figure 3.5 we can see that under each lighting condition, the most variability of the colour distribution is approximately along the Y axis. Also from Figure 3.6 we can see that as lighting condition varies, the variation of colour distribution on UV space is very small.

As Figure 3.6 shows, the colour distribution in UV space has very small variation even when the illumination goes through a dramatic change; therefore it is tempting to define a colour classifier only in UV space. In fact, our preliminary experiments show that under an illumination with mild brightness, such a colour classifier defined on UV plane is capable of accurately classifying colours despite the illumination variation within a certain range. Such a colour classifier defined on UV space is indeed very robust and accurate under dynamic lighting, but it has the inherited problem of over-classification. This is especially true when the illumination brightness is increased to a certain point. Under such a condition, the classifier defined on UV space generally over-classifies the image and generates a large number of false positives. This is due to the fact that when the illumination brightness increases, the surface reflection component grows to be significantly distinctive from the body reflection component. As shown in Figure 3.5, as the surface reflection component

does not align with the Y axis well, the 2D colour model built with the projection on UV plane of both the body reflectance component and the surface reflectance component will be an over-generalized classifier.

In our algorithm we use YUV 3D space. A colour classifier defined in YUV space is more accurate as a classifier defined in UV space, because a classifier in YUV space incorporates the information of luminance, while a classifier in UV space ignores the luminance completely. Having said this, a classifier in YUV space has the benefit of having less over-classification. In our case, over-classification needs to be as low as possible. Although most of the over-classified pixels can be filtered out with spatial constraints imposed after colour classification, the large number of false positives still could cause “phantom robot” or “phantom ball” in the output of the vision system, which may lead to making mistakes in game strategy. Meanwhile, we can always compensate the robustness by updating the classifier as lighting changes. Our experiments illustrates that a classifier defined on YUV space has satisfactory performance.

Although YUV 3D space is used for colour modeling, we build the GMM only on three colour planes, which are YU, YV, and UV planes. The colour distribution in 3D YUV space is projected onto these three planes, and the projected distributions on these planes are used to build the GMM on the corresponding plane. In other words, instead of using a single GMM in 3D colour space to model the colour distribution, we use three separate GMMs on three colour planes to define a colour model. The reason is mainly computational. It is not feasible to build the parametric model in 3D space in a real-time video sequence. Also, in order to maintain efficiency, we need to convert the GMM to a lookup table after evaluating the probability of the pixel values in each of the table entries. Evaluating the probability in 3D space with a GMM

is time-consuming, and 3D lookup table requires large memory size. While reducing the dimensionality of colour space to 2D can satisfy building model in real-time, it also reduces the memory storage of the lookup table converted from the GMM. We make the assumption that in our problem domain, two Gaussian components in 3D space are also differentiable after being projected onto 2D plane. Based on this assumption, modeling in 2D space is a close approximation of modeling in 3D space. With more computational power, directly modeling in 3D would be more accurate.

3.3 Colour Adaptation

The adaptation scheme used in our system is a simple exponentially decaying function. We avoid making linear assumption of the colour evolution. As discussed earlier, the linear transformation is applicable for modeling the evolution of the colour distribution from an approximate Lambertian surface, while in our case the surface reflectance and the body reflectance together can not be described with a linear transformation. Instead we apply a simple exponentially decaying function to update the colour model using the GMM derived at each time step.

Suppose at time step t^k , a number of sample pixels $\{p_i^k, i = 1 \dots n\}$ are collected from the image frame, and a measured GMM, GMM_m^k , is built for this colour using these pixels. Also we have a current colour model estimated at time step $k - 1$, as GMM_e^{k-1} . Now we need to use the previously estimated model GMM_e^{k-1} and the currently measured model GMM_m^k to derive the currently estimated model GMM_e^k .

Since the GMM is actually built on 3 2D projection planes, we can convert each GMM to a probability distribution table in the corresponding projection plane. i.e., for each colour we have three probability tables: P_{UV} , P_{YU} . and

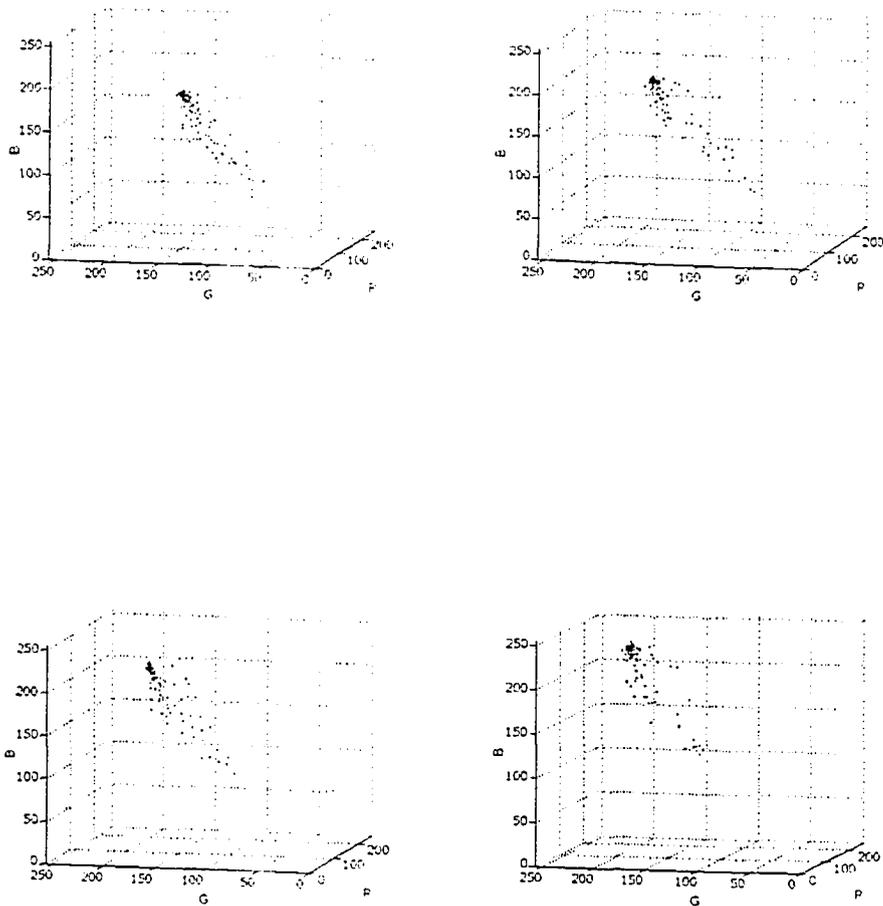


Figure 3.7: The colour distribution of a blue colour patch in RGB space

P_{YV} . Each of these tables is of dimension 256×256 , and the entry contains the probability value of this colour being evaluated by the GMM.

Suppose we have a decay factor of α , the indexes of the probability table are $i = 1 \dots 256, j = 1 \dots 256$, our exponentially decaying mechanism will update each entry of the probability tables as:

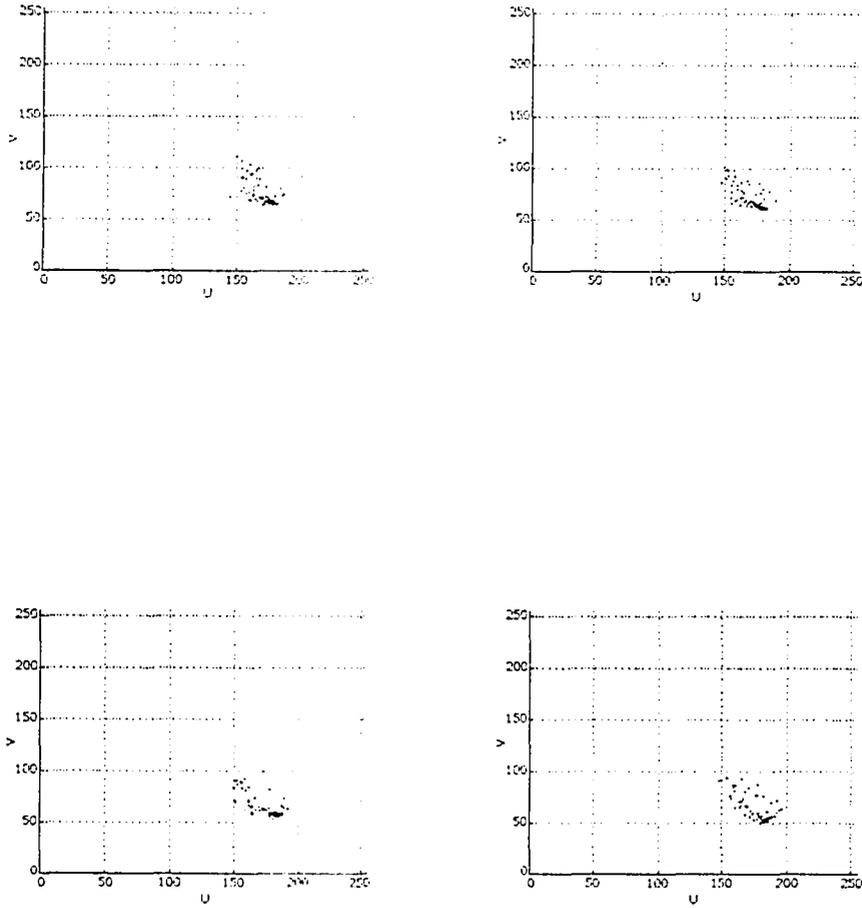


Figure 3.8: The colour distribution of a blue colour patch in UV space

$$\begin{aligned}
 P_{UV}^t(i,j) &= \alpha P_{UV}^{t-1}(i,j) + (1 - \alpha) P_{UV}(i,j) \\
 P_{YU}^t(i,j) &= \alpha P_{YU}^{t-1}(i,j) + (1 - \alpha) P_{YU}(i,j) \\
 P_{YV}^t(i,j) &= \alpha P_{YV}^{t-1}(i,j) + (1 - \alpha) P_{YV}(i,j)
 \end{aligned} \tag{3.16}$$

The decay factor α represents the extent of the influence from the previously estimated colour model on the current colour model. The range of α is within $[0, 1]$, 0 means the previous colour model is totally discarded when

update the colour model, and the measured colour model entirely contributes to the estimated current model; 1 means that the previous colour model is kept as the estimated current model, the measured colour model is entirely discarded.

It should be noted that with ground truth unknown, a GMM built from the pixels labeled at each time step is only a weak classifier. The error introduced into the adaptation includes both false positive and false negative. Our algorithm reduces the error by applying spatial constraints on the colour blobs.

As illumination changes on the field, the classification rate will drop if we use the colour model from the last time step. This introduces false negative, which are the pixels belonging to the interesting colour but not being classified into this colour class. As the labeled pixels are used as samples to update the colour model, the decreasing of the number of the labeled pixels will cause the colour distribution to be only a subset of the colour model and biased toward the original model. Based on the observation that target colour blobs are closed regions in both image and colour space, a region growing is done in order to reduce false negative. Once the classified blob size is below a certain threshold, the pixels classified with the current lookup table are used as seeds to grow region on their neighboring pixels according to the pixel distance in the colour space. The classified pixels and the grown pixels together are selected as samples to build the GMM. The threshold for region growing is adjusted automatically by the system according to the size of the blobs being identified.

A common problem which exists in an adaptive algorithm is the possibility of converging to a false target. The adaptation is done based on the labeled data collected during the unsupervised tracking. While the ground truth is unknown, the labeled data might include false positive as noise colours being

classified as the target colour. The noise colour could attract the colour model to drift away from the target colour entirely if the noise colour gets more weight during adaptation, which will eventually lead to the failure of tracking. In our algorithm, a spatial constraint of the colour blobs is applied to reduce false positive. Because most of the false positive are from the boundary of the colour blob, only the pixels within a certain range around the center are selected as samples to update the colour model.

Chapter 4

System Description

Our adaptive colour classification algorithm is implemented in the colour vision system for RoboCup team Team Canuck. The vision system uses an IEEE 1394 (Firewire) Dragonfly camera and a normal Linux PC. The camera is made by Point Grey Research Inc., and it features a single 1/3" progressive scan CCD. The PC has an off-the-shelf Firewire capture card, and AMD Athlon XP 1700+ processor with 512 MB of RAM. We use the camera to sample 640×480 24 bit RGB colour images at 30 fps into the PC. Since the camera only provides a monochrome image in raw Bayer format, i.e. it only provides one of the RGB components at each pixel, simple convolution matrices are used to generate RGB values at each pixel from the camera output [41]:

$$\begin{aligned} M_1 &= \begin{bmatrix} \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \end{bmatrix} & M_2 &= \begin{bmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{bmatrix} \\ M_3 &= \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \end{bmatrix} & M_4 &= \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (4.1)$$

M_1 and M_2 are used at blue (red) pixels to fill in the missing red (blue) and green values. M_3 and M_4 are used at green pixels to fill in missing red and blue values [41].

The vision system uses a graphical user interface (GUI) to display the

continuous video sequence from the camera, as shown in Figure 4.1. The GUI was originally designed by Team Canuck for their previous vision system, and has been modified for the current adaptive system. The main functionality of the GUI was to provide a visual debugging mechanism to calibrate camera and to set parameters for the vision system. Either the classified pixels, or the classified colour blobs, or the identified robots can be shown on the video display window within the GUI. As shown in Figure 4.7, each identified robot is marked on its top-right with a number which corresponds to the id of this robot, and the classified pixels are drawn in a unique colour on the top of the original pixels, also the classified center colour blobs are marked with a small red square. This provides to the user a very intuitive way for visually inspecting the classification result on the video sequence, it also provides to the user an intuitive way for tuning the parameters of the vision system since the result of classification is directly displayed on the video display.

The previous system requires an off-line manual colour calibration process, which uses a stand-alone software module, as shown in Figure 4.2. Firstly, an image frame is captured from the video sequence and then loaded into the manual calibration tool. With the assistance of the interactive function of this tool, the user can select colour pixels of the same colour class by hand from the captured image, and then the selected pixels are projected as binary masks on three 2D planes in the RGB colour space. In Figure 4.2, these three 2D binary masks are shown in the three windows on the left hand side of the GUI. The calibrated binary masks are applied onto the captured image to examine the accuracy of the calibration. Each pixel which falls into three binary masks is labeled with this colour class. Figure 4.3 shows the classification results after the binary masks are applied on the captured image. The labeled pixels are drawn as black, and the background pixels are drawn as white. After colour

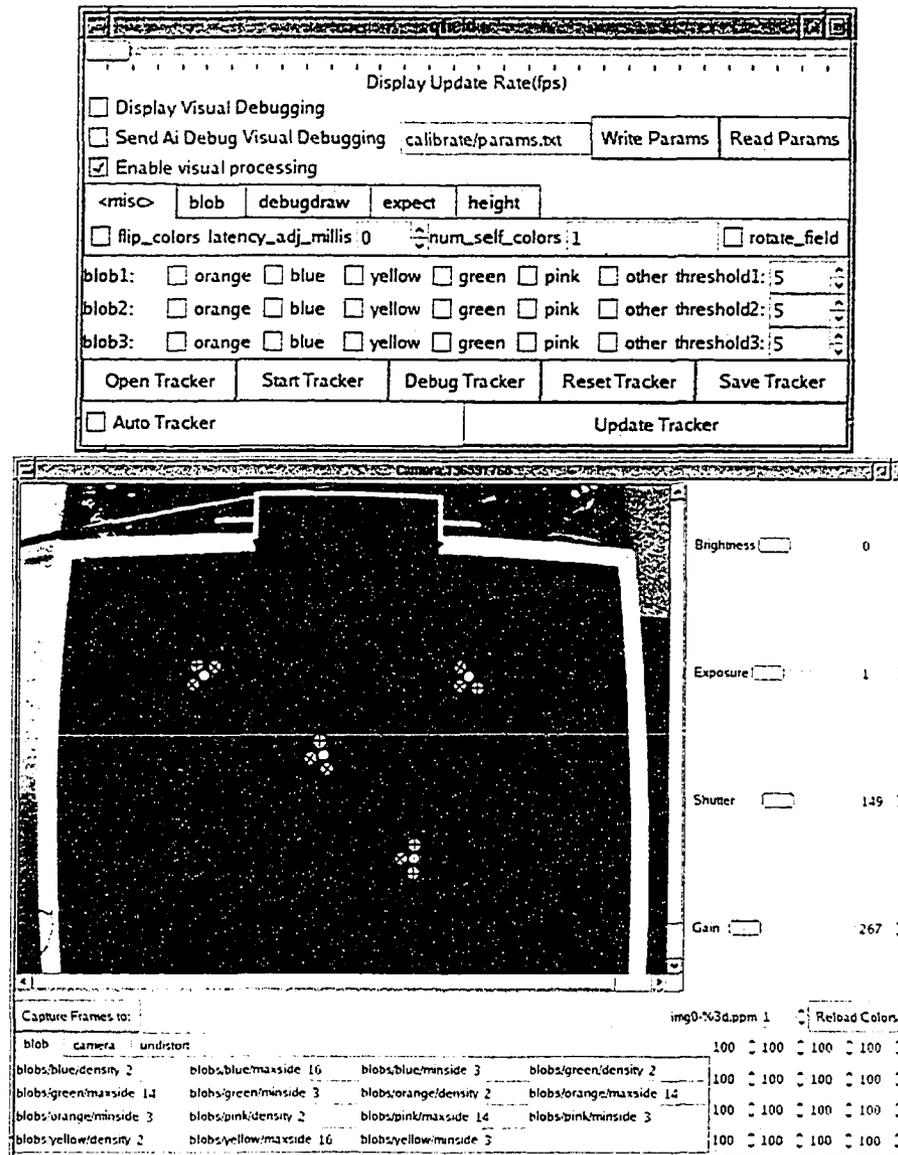


Figure 4.1: System GUI. Pink, green, and blue blobs are illustrated with x, +, and circle.

calibration is done, these binary masks are saved to mask files and then the vision system loads the binary mask files as LUTs for colour classification. The current system replaces this stand-alone colour calibration tool and integrates the colour calibration functionality within the vision system. The LUTs are now calibrated on-the-fly as part of the system functionality. Instead of taking

approximately an hour to build the LUT using the previous system, it now takes only a few minutes to calibrate accurate LUTs on the entire field. For the current vision system, the GUI now includes functionalities for colour calibration, camera calibration, and the vision parameter tuning.

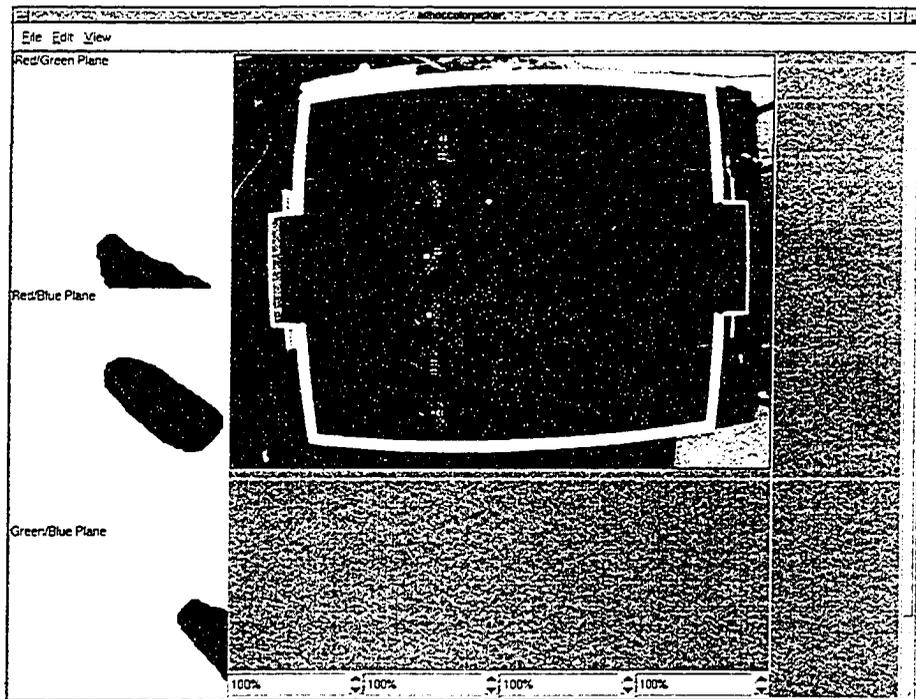


Figure 4.2: Adhoc-colourpicker GUI calibration step 1. Pink, green, blue, and yellow blobs are illustrated with \times , $+$, circle, and square.

The system works in two steps. The first step is off-line colour calibration, and the second step is on-line adaptive colour classification. Both are described in detail in this chapter.

4.1 Off-line Colour Calibration

The off-line colour calibration step extracts the colour pixels of the target colour from each frame of the video sequence, and uses these sample pixels to build a colour model which incorporates spatial variation of illumination

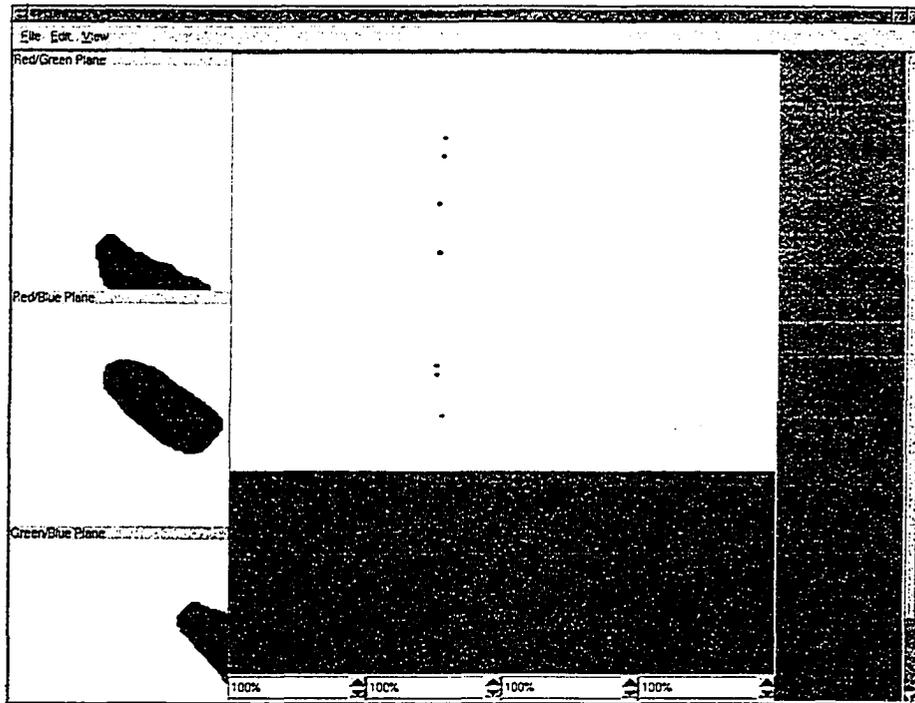


Figure 4.3: Adhoc-colourpicker GUI calibration step 2

on the field. This step is under the supervision of a human operator, and the finalized model is used as an accurate colour model to initialize the system.

To start the calibration step, user selects pixels of the interested colours from the colour blobs in an image using interactive tool provided by the system. Once the pixels are selected, a simple colour tracker starts to track the movement of the object which the selected pixels belong to. The tracker selects an image region as a tracking window which includes the object, within the tracking window a region growing starts from the selected pixels as seeds to include neighboring pixels of the same colour. The pixels being selected by region growing form blobs, and the locations of the blobs are used as a cue to update the position of the tracking window in the consequent image frames, so that the tracker follows the movement of the object over the field. Figure 4.5 illustrates how the tracker follows the object of the selected colour blobs,

where the red square drawn around the top-left robot represents the tracking window of the tracker. For each image frame, the tracker collects all the pixels selected by region growing inside the tracking window as the sample pixels of the interested colour. As the robot moves around over the field, the set of sample pixels can incorporate the variation of the pixel values due to spatial variation of illumination. In summary, during the auto-calibration step, the sample pixels are collected by a tracker while it follows the movement of the object over the entire field, so that it samples the pixels of a certain colour under the illumination of spatial variation.

This data collecting process stops when the robot approximately goes through representative locations on the field, especially the places with significant different illuminations. The sample pixels collected during this step are then used to derive the GMM of two components using a standard EM algorithm. The derived GMM is used by the vision system as colour model for classifying this colour. With the GMM of the colour class, the classification for each pixel can be done by calculating the probability of this pixel with the model. As this calculated probability represents the probability for this pixel to be within this colour class, a simple thresholding method will classify this pixel as within the colour class or not. For efficiency, instead of evaluating each pixel with the GMM, each GMM is used to derive the probability distribution on each of the three 2D projection planes, which are YU, YV, and UV plane. After the probability distribution on each plane is built, this probability distribution is converted to a lookup table after thresholding the probability at each entry. The classification for each pixel is done by checking the entry of this pixel in the three lookup tables.

As being mentioned above, the previous vision system provides a debugging mechanism which visualizes the classified pixels and the classified colour blobs

in each image frame of the video sequence. Also, the robot being identified with all its colour blobs located is visualized with its number shown nearby. As shown in Figure 4.7, each of the robots on the field has colour blobs of three different colours, blue, pink, and green. The debugging functionality visualizes these colours with three different colours to illustrate the result of colour classification, and each robot has its identification number displayed next to it. This provides a convenient way to visually inspect the accuracy and the completeness of the colour model for classifying a certain colour. During this auto-calibration step, the user can examine the calibrated colour table by using this debugging functionality to inspect the result of the classification on video sequence. When all the robots in the field are correctly identified while they are moving around over the field, the derived colour model is accurate and complete enough. At this point, the user can stop the colour calibration step. Otherwise, the user can repeat this process and refine the colour model till it is acceptable.

4.2 On-line Adaptive Colour Classification

When a game starts, the pre-calibrated colour models are used as the ground truth by vision system for colour classification. Meanwhile, an adaptation mechanism is imposed to update the colour model in real-time in order to adapt to dynamic lighting.

For each image frame fed into the vision system, a colour classification is done for all the pixels in a scan line order using the current LUTs. This process labels each pixel with a colour class, either one of the interesting colours, or the background. The pixels of each of the interesting colour classes are extracted from the image, and then grouped together into a region if they are spatially connected. At this step, a region growing is applied using each of these labeled

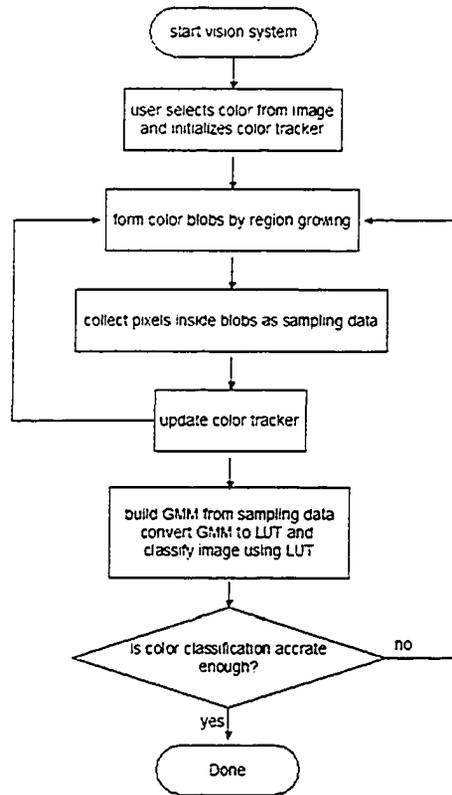


Figure 4.4: Offline colour calibration

pixels as seed to include the neighboring pixels, which has pixel values close in colour space but are not being classified within the colour class with the current colour model. The reason why these pixels with the same colour are not picked up by our current colour table is either that the sample pixels collected during the calibration step are not a complete representation of the entire field, or that the illumination on the field has gone through a temporal change such that some of the pixel values fall outside of the calibrated colour model. In either way, we have a number of false negatives due to possibly spatial variation or the temporal variation of the illumination. The purpose of the region growing is to reduce false negatives. We want to ensure that the colour pixels which shifts beyond the current colour model due to illumination change are also being correctly labeled. In this way, we can compensate the

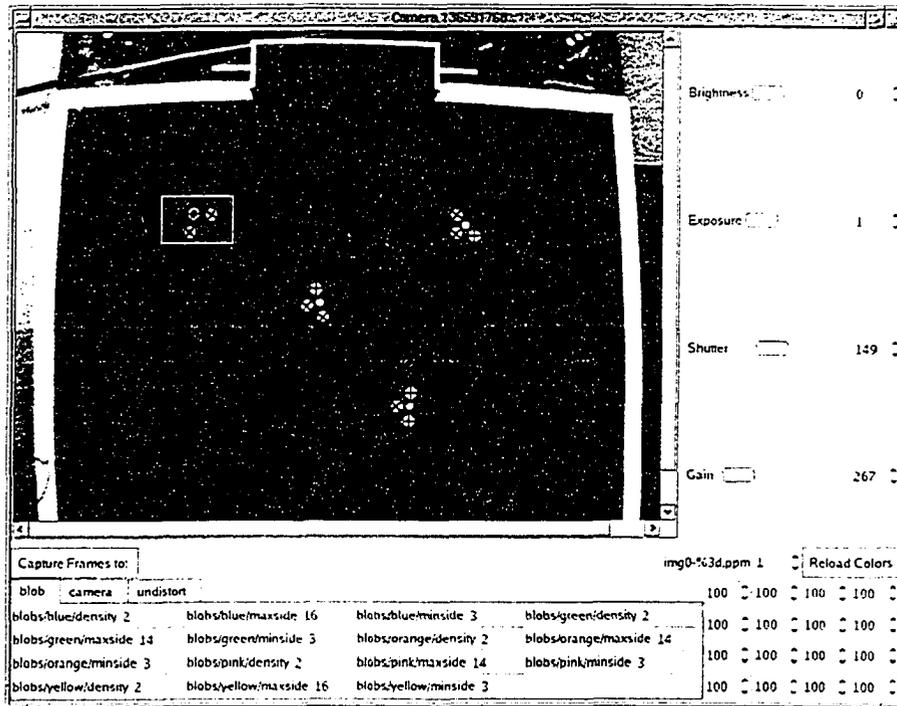


Figure 4.5: Initialized colour blob. Pink, green, and blue blobs are illustrated with \times , $+$, and circle.

inaccuracy of our current colour model by incorporating the pixel values due to spatial or temporal variation of illumination. Because the colour model is updated with the labeled pixels, we need to include the false negatives to update our colour model to adapt to the current environment. If we only use pixels being labeled with the current colour model, since these pixels are only a subset of the ground truth if there is any illumination change, the adaptation of the colour model will lead to a subset of the previous model and will be biased toward the previous model.

After all the colour pixels of the interesting colours are classified in an image, the labeled pixels and their neighbors of the same colour class are connected to form a number of colour blobs. Usually the colour blobs are of different size and shape, some of which are noise due to the background noise or over-classification of the inaccurate colour model. The noisy blobs

are filtered by applying a set of spatial constraints, such as the height and the width of each blob, the density of the colour pixels within each blob, etc. These spatial constraints are based on our a priori knowledge of the size and shape of the target blobs used in this application. Now the filtered blobs are the candidates of potential markers of the robots and the ball. After being evaluated with the sets of spatial constraints described above, these candidate blobs are recognized as belonging to either a robot or the ball. By now the robots and the ball are located and identified.

For each of the identified robots and the ball, their colour blobs are used as labeled samples to derive the GMM as the measured colour model at the current time step. Since there are a number of false positives within the labeled pixels, to reduce the influence of the false positives on the measured model, for each blob only pixels in the center region within a certain radius are collected as sample pixels to build GMM. This is based on our observation that the boundary pixels of each blob are affected by the colour region surrounding the blob due to camera noise. These pixels are likely to attract colour model toward a false target.

The evaluation of the GMM from collected labeled pixels is done for every 20 frames. It is unnecessary to reevaluate the GMM for each frame unless the illumination condition goes through a rapid change, in which case reevaluation and update of the colour model needs to be done for each frame. In our application the rapid change of illumination rarely happens. The first benefit of reevaluating the GMM after a time interval is efficiency. Another benefit is that since the robots are moving around over the field, a number of frames can provide a more complete set of sample pixels which capture the spatial variation of illumination; therefore the collected pixels during this time interval will incorporate the spatial variation of the entire field. After a GMM is built

for each colour, its Gaussian components are used to evaluate the probability distribution on UV, YU, and YV plane. After the probability distribution on each plane is used to update the current distribution table, it is converted to a lookup table and used for colour classification at the next frame.

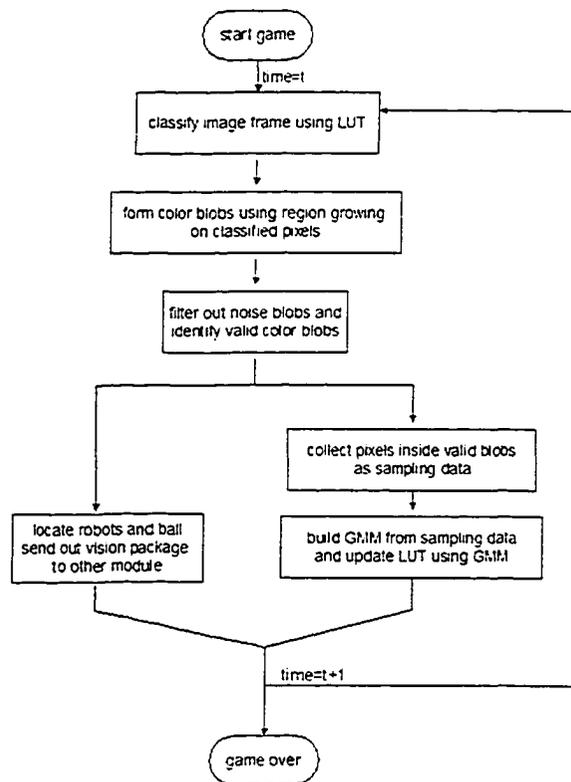


Figure 4.6: Online adaptive colour classification

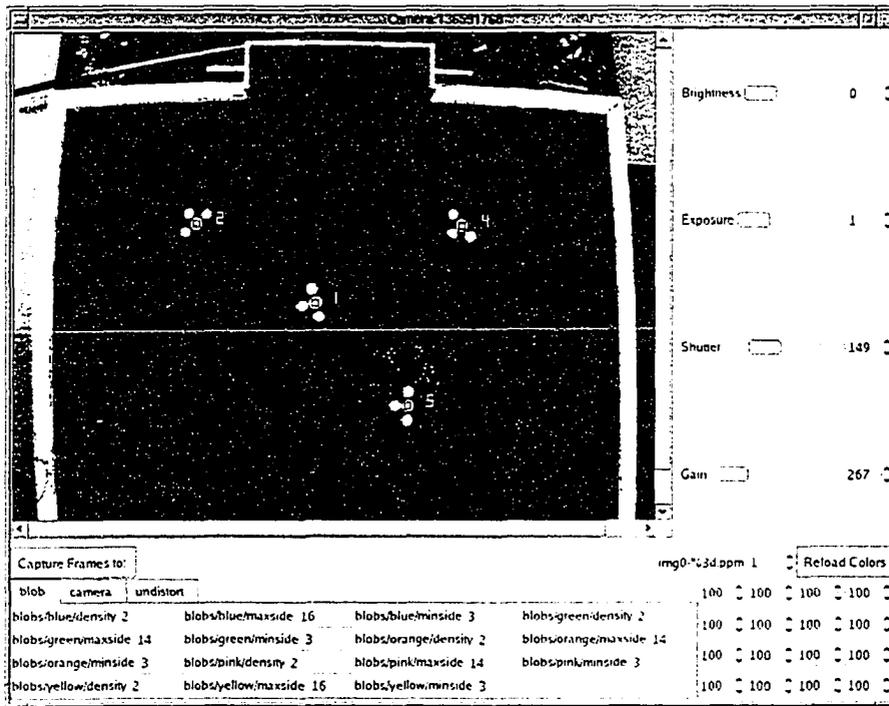


Figure 4.7: Colour classification using LUT built from initialized colours. The classified colour pixels are illustrated in white.

Chapter 5

Results

Experiments were carried out to test the efficiency and the accuracy of the adaptive colour classification algorithm. The vision system with adaptive colour classification can process approximately 30 frames/second, which is of satisfactory performance for a real-time vision application. It should be noted that our vision system does RGB to YUV transformation of each image frame with software since the camera only provides RGB image. This transformation process costs about 15 millisecond for each frame on our system, which is about 50% of the total time for processing one frame. Using a camera with YUV image input can reduce this time and will make our algorithm more efficient.

The adaptive colour classification algorithm was tested under laboratory environment. The illumination was gradually changed by varying the room lighting between dim and bright. The illumination on the field was measured with a photographic meter MINOLTA auto meter VF. The range of the average illumination measured on the field was from 70 Lux to 256 Lux as the room lighting was varied from dim to bright. The spatial illumination variation on the field was between 40 Lux and 80 Lux at the lowest room lighting, and between 135 and 320 Lux at the highest room lighting. Under this experiment condition, all the robots were correctly identified using the adaptive colour

classification algorithm. Figure 5.1 shows the correctly classified pixels from the 460 frames of video sequence using adaptive classification and non-adaptive classification, under the condition that the illumination was gradually changed between 70 Lux and 256 Lux. It shows that with adaptive classification the number of the correctly classified pixels keep approximately constant when the illumination changes, while the non-adaptive classification fails soon after the illumination changes. Figure 5.2 shows the illumination on the field varies between 70 Lux and 256 Lux during these 460 frames of video sequence.

A series of comparison experiments were carried out to examine the accuracy of our algorithm under dynamic lighting. The first set of experiments compares the classification accuracy between adaptive algorithm and non-adaptive algorithm under dynamic illumination. The results show the adaptive algorithm is capable of handling illumination change through a broad range while the non-adaptive algorithm fails very quickly after illumination changes. The second set of experiments compares the classification accuracy between using RGB space and YUV space for our adaptive algorithm under dynamic lighting. The results show that adaptive colour classification using YUV space is more capable of handling illumination change in a wide range than using RGB space. The third set of experiments compares the classification accuracy between using only diffuse component vs using both diffuse and specular component for our adaptive algorithm under dynamic lighting. The results show that adaptive colour classification using both diffuse and specular component is more capable of handling a wide range of illumination change than only using diffuse component. This chapter describes these experiments in detail.

We use the F-Measure as the criterion for measuring the classification accuracy in order to compare the performance of two algorithms. The F-Measure was first introduced by Rijsbergen [58], and it has been used widely to evaluate

the performance of a classifier. We use a weighted version of the F-Measure which combines weighted recall r and weighted precision p , as shown in the following form [47]:

$$F_{\alpha}(r, p) = \frac{(\alpha + 1)rp}{r + \alpha p} \quad (5.1)$$

where r has a weight of $\alpha \in (0, +\infty)$ and p has a weight of 1. r is defined as true positive rate, i.e, the proportion of positive cases that were correctly identified. p is defined as the proportion of the predicted positive cases that were correct. We assign $\alpha = 0.5$ so that r has more weight than p . This is because recall is more of a critical importance in our application.

In order to control the experimental condition to be the exactly same for the purpose of comparing these two algorithms, we simulate the variation of the illumination condition by changing the camera gain control. Camera gain control represents the amount of the signal amplification of the camera. By increasing a camera's gain control, it will boost the contrast and effectively increase the brightness of the image. Through changing the gain control of the camera, we get an approximate simulation of brightness change in the environment. Figure 5.3 demonstrates how the different colors vary due to the actual illumination change. Figure 5.4 demonstrates how the colors vary due to the simulated illumination change by adjusting the camera gain. These two figures together support that using camera gain to simulate illumination variation can be a reasonable approximation of the actual illumination change within a certain range. It is observed that when gain value increases in the range between 200 and 500, the color models evolves in the color space approximately along the luminance axis. When the gain value reaches approximately higher than 500, colors become saturated and adaptive modeling can not accurately model the color distribution. The situation when the gain value gets

too high is beyond the limit of our color model; therefore it is not considered in the comparison experiments. For our comparison experiments, we collected data during the gain value changes between 200 and 500.

5.1 Adaptive vs Non-Adaptive

Although the adaptive colour classification algorithm does not impose heavy computational burden on our system, it still costs more CPU time than the non-adaptive algorithm which only uses a static colour model, due to its effort of evaluating the current colour distribution and refining the current colour model. So the first question we need to answer is: how much can we gain from the extra effort of adaptive colour classification?

Our assumption is that adaptive algorithm is more robust to dynamic environment than non-adaptive algorithm; therefore it must have a better accuracy with respect to illumination change compared with a non-adaptive algorithm. To verify this assumption, we set up an experiment to compare the accuracy of colour classification between adaptive vs non-adaptive colour classification algorithms.

Figures 5.5 – 5.8 show the results from the experiments which compare the classification results of the adaptive algorithm against the non-adaptive algorithm under dynamic lighting. The gain value, which corresponds to the brightness of the image, is the x axis in all these figures. For each image frame, we record the total number of pixels being classified as within a colour class, the total number of pixels being correctly classified as within a colour class, the total number of blobs being correctly classified as within a colour class, and the total number of robots being correctly identified. In Figure 5.5, the number of robots being correctly identified is plotted. In Figure 5.6, the classification accuracy, which is computed with the F-Measure, is plotted.

Also, the number of the correctly classified pixels is plotted in Figure 5.7, and the number of the correctly classified blobs is plotted in Figure 5.8.

Figure 5.5 shows that with adaptive colour classification the robots are all identified through a wide range of illumination change. With non-adaptive colour classification, the number of the identified robots drops right after a mild illumination change and reaches zero very quickly. This comparison experiment demonstrates the practical significance of the proposed adaptive algorithm in the colour vision system, i.e, all robots can be correctly identified under lighting changes.

Figure 5.6 shows that with adaptive colour classification the score of F-Measure keeps approximately 1 through a wide range of illumination change. With non-adaptive colour classification, the score drops after a mild illumination change, and reaches zero soon. It is observed that for the green colour the F-score drops when the brightness of the image reaches the point of being very high. This is because that once the brightness of the image becomes very high, the green colour and the background colour are all saturated and not differentiable any more.

Figure 5.7 shows the absolute number of the pixels which are correctly classified as within a colour class, with adaptive colour classification and non-adaptive colour classification respectively. With non-adaptive colour classification, the number of the correctly classified pixels drops soon after the illumination changes, while with adaptive colour classification, the number of correctly classified pixels remain non-decreasing through this process. The correctness of the classified pixels are based on comparison between the classified results with the manually-classified results.

Figure 5.8 shows that all the colour blobs are correctly classified with adaptive colour classification through the process of illumination change, while with

the non-adaptive colour classification the number of blobs being identified drops soon after the illumination changes and reaches zero quickly.

It is observed that the adaptive colour classification can adapt to the lighting change and remain a high classification accuracy, and the robots on the field are all identified through the lighting variation till the brightness reaches a certain point that colours are not differentiable any more; meanwhile the non-adaptive colour system fails right about the lighting changes.

5.2 RGB vs YUV

We have carried out comparison experiment to evaluate the classification accuracy of the adaptive colour classification between RGB colour space and YUV colour space. Figure 5.9 – 5.10 show the results from the experiments which compare the classification accuracy of the adaptive colour system using YUV colour space against the adaptive vision system using RGB colour space under dynamic lighting. The lighting change is simulated by increasing the camera gain. The gain value, as before, which corresponds to the brightness of the image, is the x axis in all these figures. For each image frame, we also record the total number of pixels being classified as within a colour class, the total number of pixels being correctly classified as within a colour class, the total number of blobs being correctly classified as within a colour class, and the total number of robots being correctly identified.

In Figure 5.9, the number of robots being correctly identified is plotted. Figure 5.9 shows that with adaptive colour classification in YUV space the robots are all identified through a wide range of illumination change. With adaptive colour classification in RGB space, the number of the identified robots drops soon after a mild illumination change and reaches zero very quickly.

In Figure 5.10, the classification accuracy represented with F-Measure is

plotted. Figure 5.10 shows that with adaptive colour classification in YUV space the F-score keeps approximately 1 during the illumination change. With adaptive colour classification in RGB space, the classification correct rate drops after a mild illumination change, and reaches zero soon. For the green colour, the classification accuracy drops when the illumination brightness reaches the point that it is not differentiable from the background green any more.

It is observed that the adaptive colour classification with YUV space can adapt to the lighting change and maintain a high classification accuracy, and the robots on the field are all identified through the lighting variation till the brightness reaches a certain threshold that colours are not differentiable with the background; meanwhile the adaptive colour system with RGB space fails right about the lighting changes.

5.3 Diffuse vs Specular and Diffuse

Figure 5.11 – 5.12 show the results from the experiments which compare the classification accuracy of the adaptive colour system using only the diffuse component and both diffuse and specular components in YUV colour space under dynamic lighting. The two components of the GMM were separated according to their Y values. Based on the observation that diffuse component has a lower luminance than specular component, we assume that the one component with a smaller Y value in YUV colour space is the diffuse component, the one component with a larger Y value is the specular component. This assumption is consistent with the “T-shape” of dichromatic reflectance property. For this group of comparison experiments, the lighting change is also simulated by increasing the camera gain. The gain value, which corresponds to the brightness of the image, is the x axis in all these figures. For each image frame, again, we record the total number of pixels being classified as within

a colour class, the total number of pixels being correctly classified as within a colour class, the total number of blobs being correctly classified as within a colour class, and the total number of robots being correctly identified.

In Figure 5.11, the number of robots being correctly identified is plotted. With adaptive classification algorithm using both diffuse and specular components, the number of correctly identified robots remains constant through a broad range of illumination brightness change, till the brightness reaches very high. With adaptive classification using only the diffuse component, the number of correctly identified robots drops right after a mild illumination change, and decreases to zero quickly.

In Figure 5.12, the classification accuracy, measured with the F-Measure, is plotted. Figure 5.12 shows that the F-scores of the adaptive classification using both components are significantly higher than the F-scores of the adaptive classification using only diffuse component through a wide range of illumination change. With adaptive colour classification using both components the classification accuracy score drops only after a broad range of illumination brightness change. With adaptive colour classification using only diffuse component, the F-score drops after a mild illumination change, and reaches zero soon.

It is observed that the adaptive colour classification of both diffuse and specular component can adapt to the lighting change and remain a high classification accuracy, and the robots on the field are all identified through the lighting variation till the brightness reaches a certain threshold that different colours are not differentiable any more. Meanwhile the adaptive colour system of only the diffuse component fails right about the lighting changes.

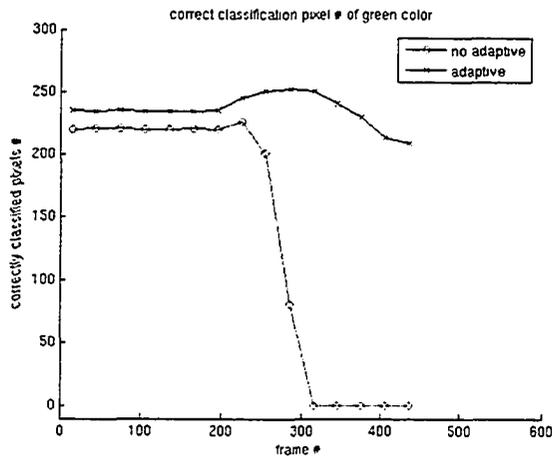
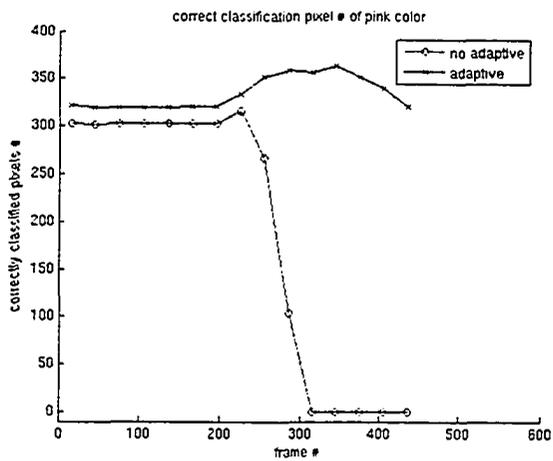
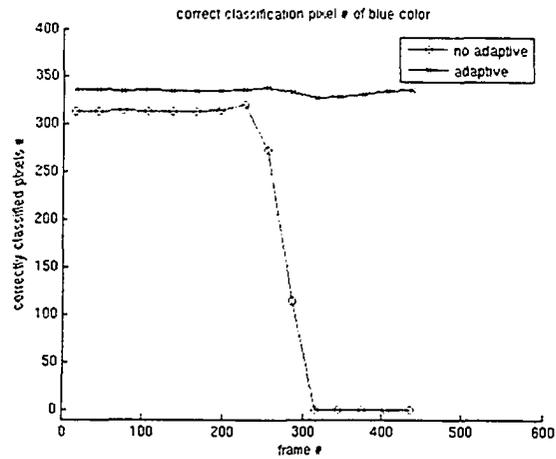


Figure 5.1: The correctly classified pixels with adaptive colour classification vs non-adaptive colour classification under gradually changing illumination from 70 Lux to 256 Lux.

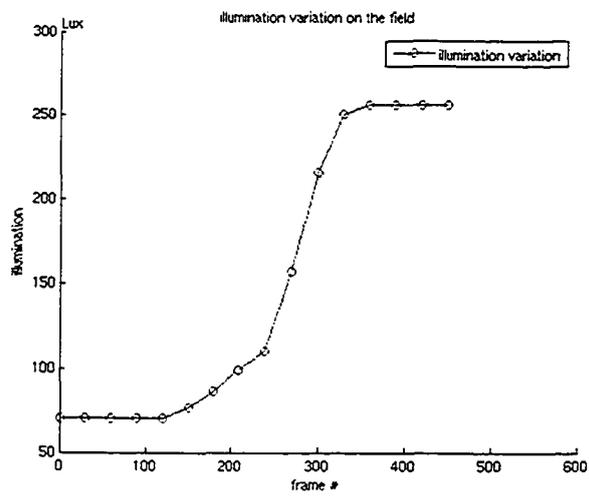


Figure 5.2: The gradual illumination variation on the field from 70 Lux to 256 Lux.

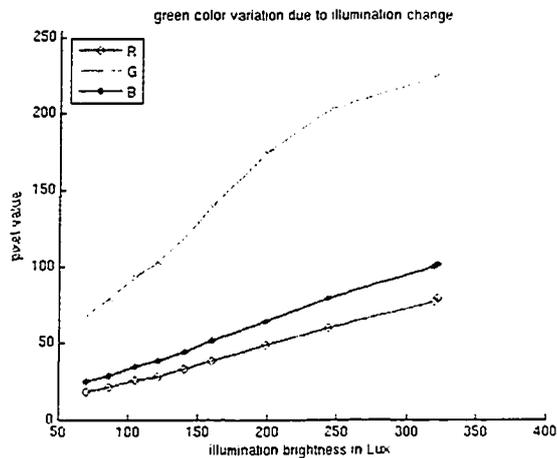
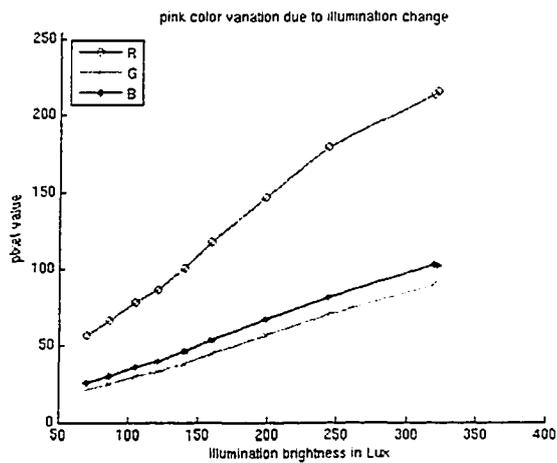
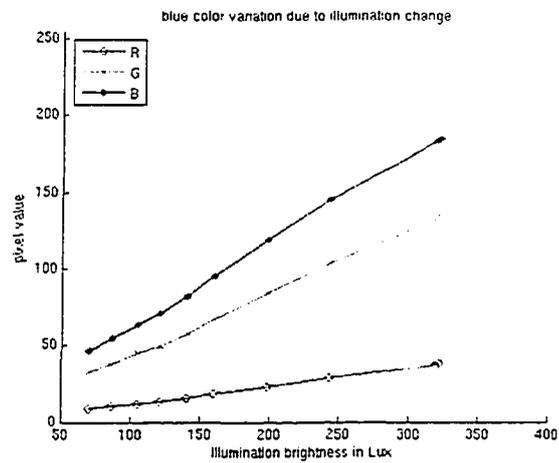


Figure 5.3: The color variation under gradually changing illumination from 70 Lux to 322 Lux.

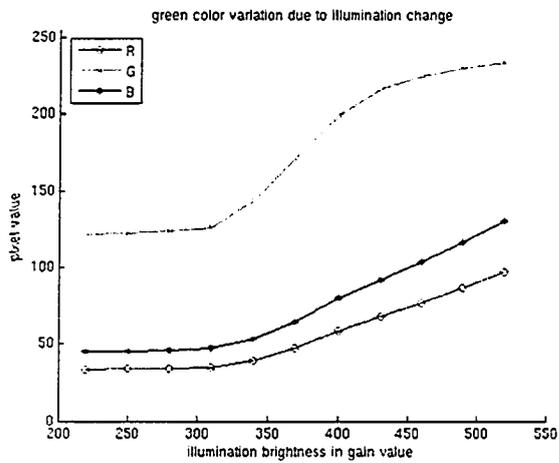
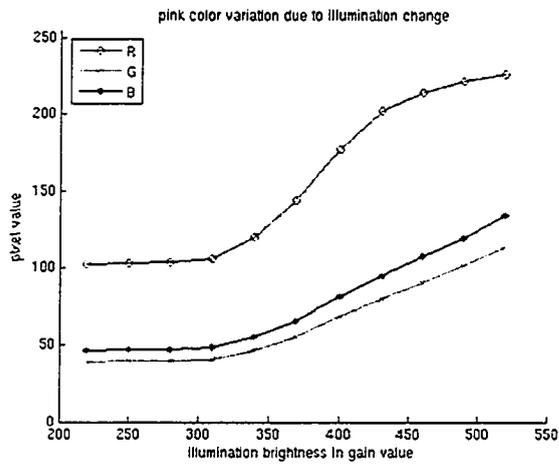
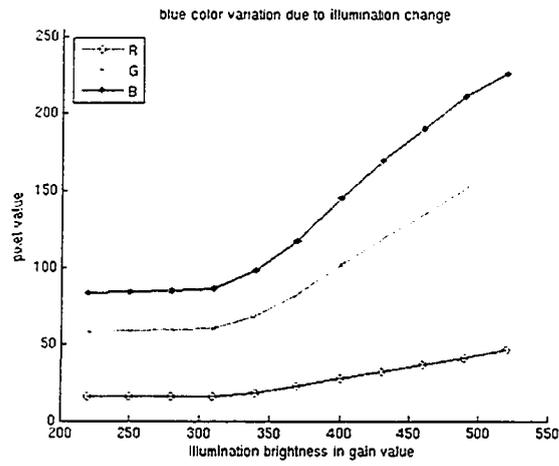


Figure 5.4: The color variation under gradually changing illumination simulated by changing the camera gain from 220 to 550.

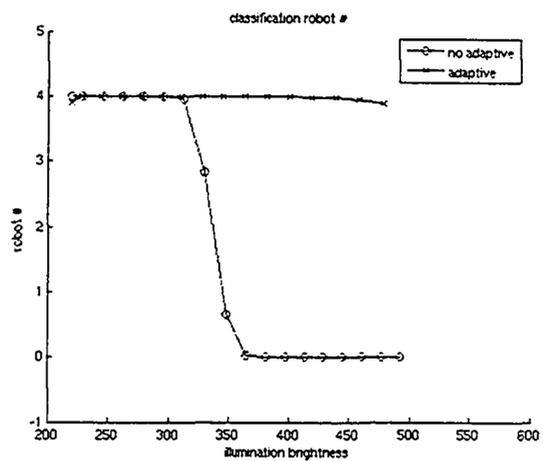


Figure 5.5: Comparison of the number of correctly identified robots between adaptive colour classification and non-adaptive colour classification under gradually increasing lighting. The total number of robots is 4.

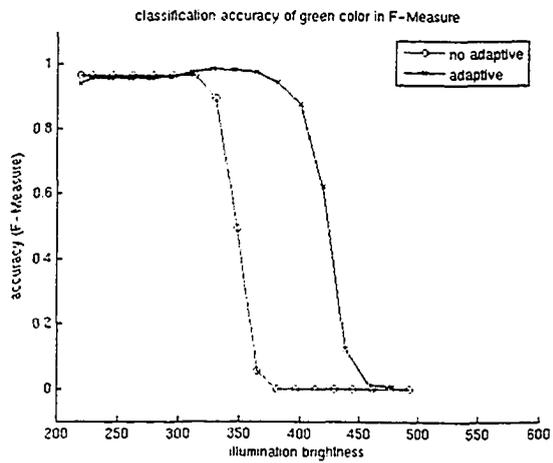
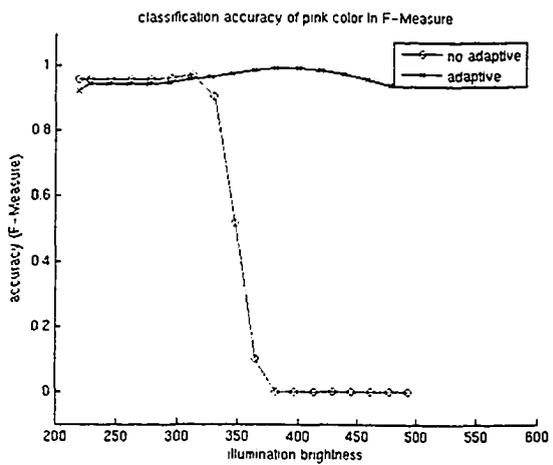
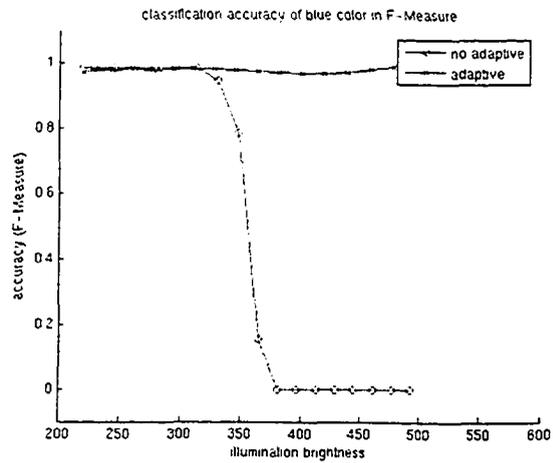


Figure 5.6: Comparison of the classification accuracy between adaptive colour classification and non-adaptive colour classification under gradually changing lighting.

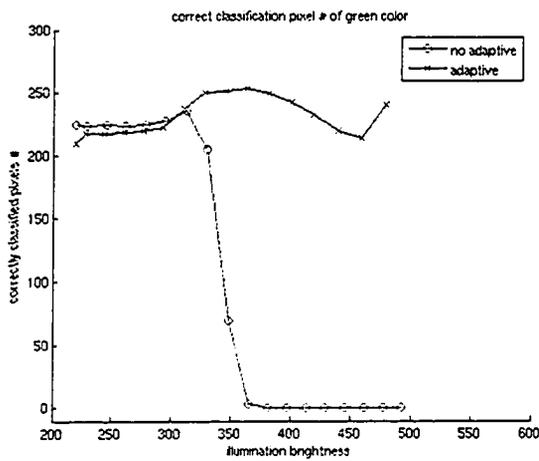
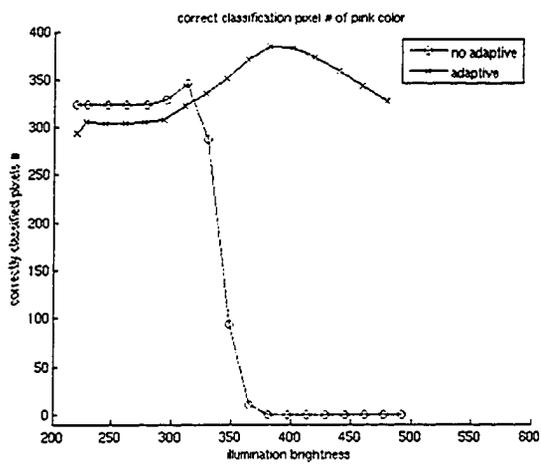
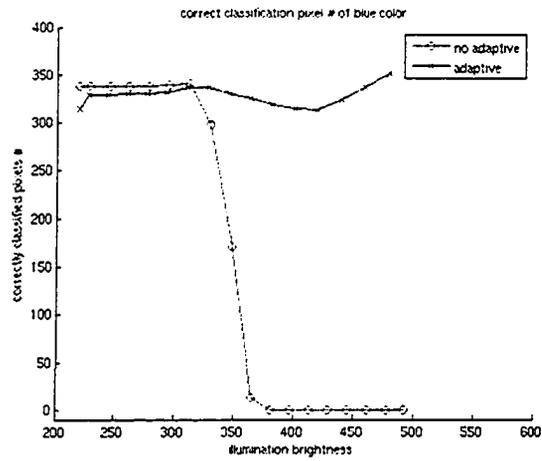


Figure 5.7: Comparison of the correctly classified pixels between adaptive colour classification and non-adaptive colour classification under gradually changing lighting.

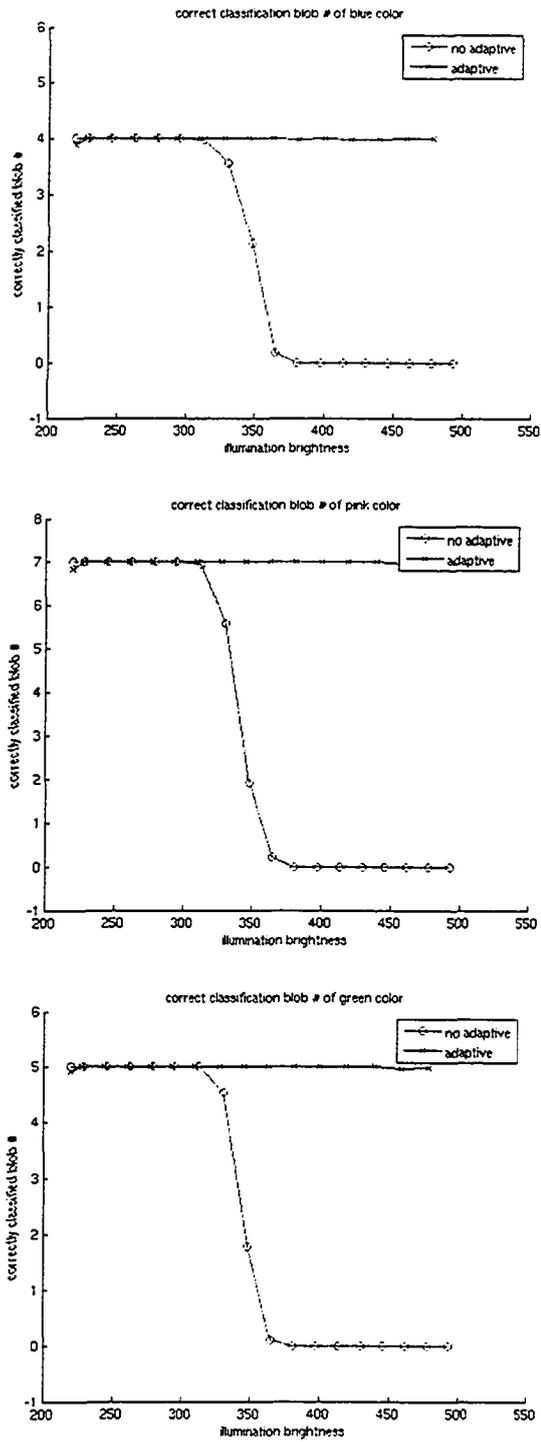


Figure 5.8: Comparison of the correctly classified colour blobs between adaptive colour classification and non-adaptive colour classification under gradually changing lighting.

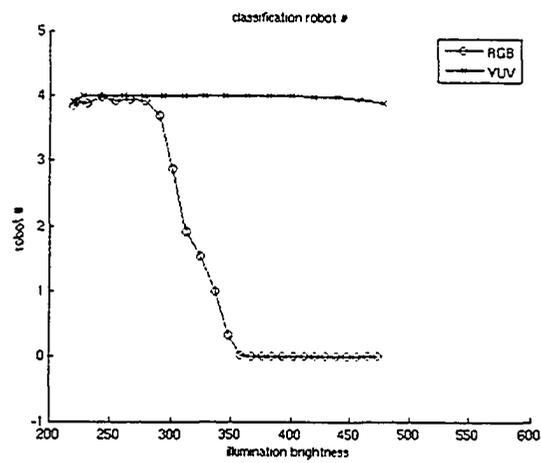


Figure 5.9: Comparison of the correctly identified robots by adaptive colour classification under gradually changing lighting between RGB and YUV colour space

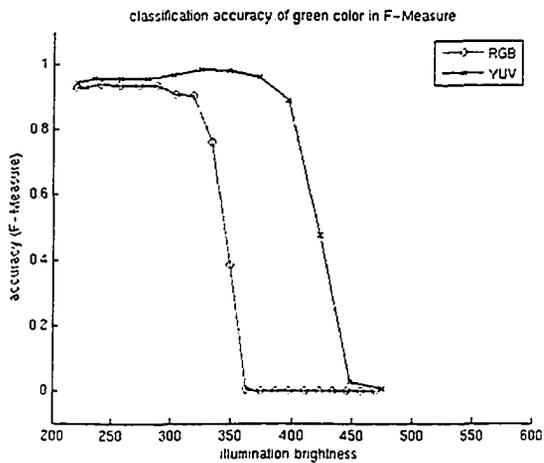
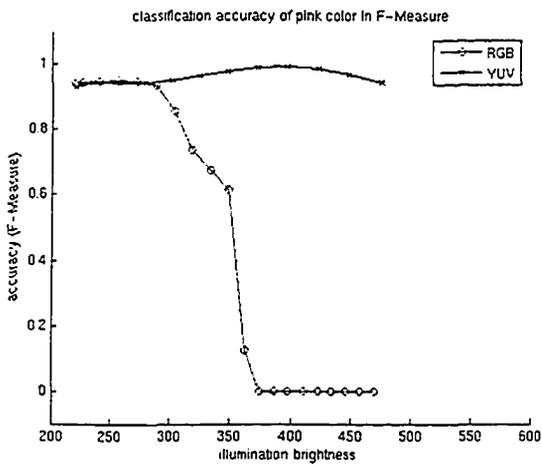
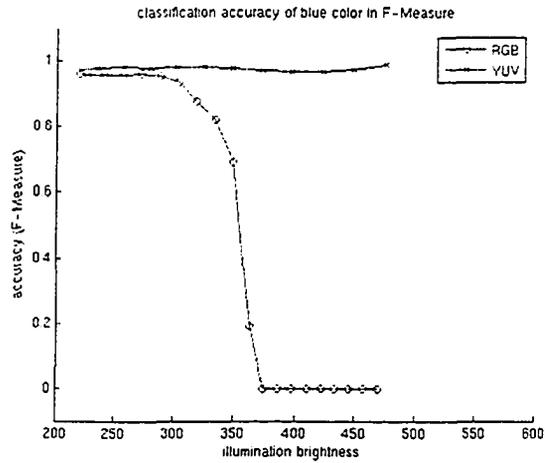


Figure 5.10: Comparison of the classification accuracy of adaptive colour classification under gradually changing lighting between RGB and YUV colour space.

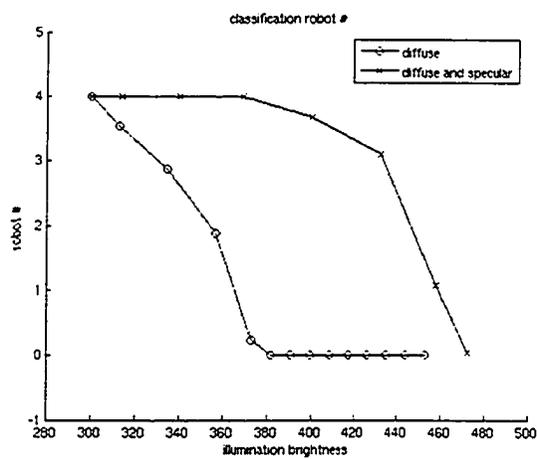


Figure 5.11: Comparison of the correctly identified robots by adaptive colour classification in YUV space under gradually changing lighting between using only the diffuse component and both diffuse and specular components

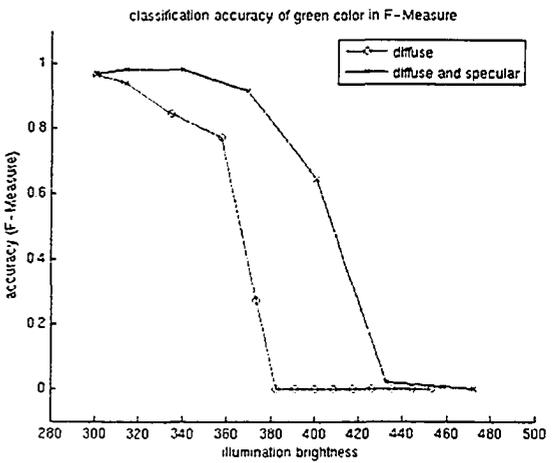
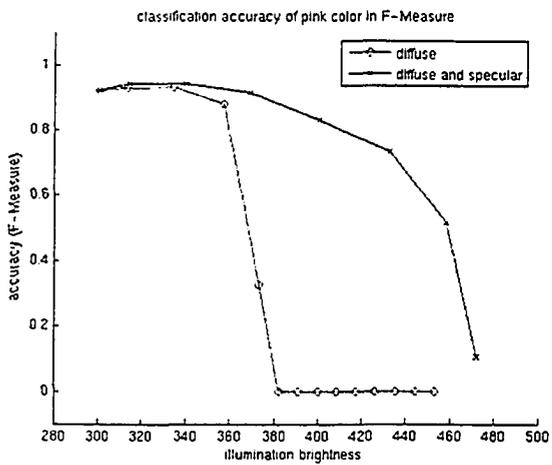
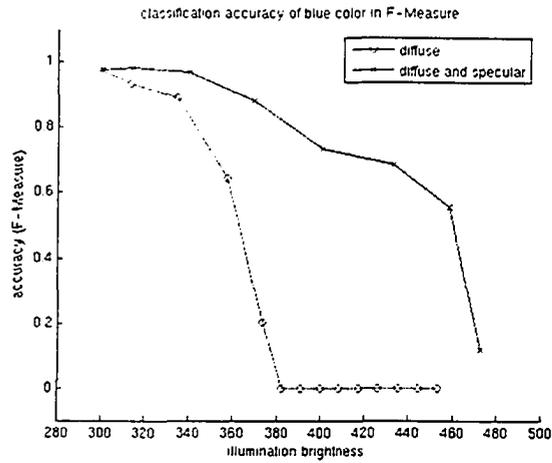


Figure 5.12: Comparison of the classification rate between using only diffuse component and using both diffuse and specular components with adaptive colour system under gradually changing lighting.

Chapter 6

Discussion and Conclusion

The adaptive colour classification algorithm is tested under gradually changing lighting condition. The GMM colour model is adapted to the illumination change, and colour classification on all interesting colours is based on the adaptive colour model. Using the proposed adaptive classification algorithm, the vision system has a satisfying performance as all the robots are located and identified correctly under a wide range of illumination variation.

Our algorithm is closely related to Raja's work, as both use GMM as colour model in an adaptive colour classification algorithm [63]. The major difference between our work and Raja's work is that Raja uses GMM to model a multi-coloured surface, while we use GMM to model a uniform-coloured dichromatic surface. Raja's work treats the specular highlights as outliers points during the modeling process, and claims that these outliers points are possibly caused by the image noise and specular highlights have little influence upon the mixture model [63]. While in our work, specular highlights are not negligible component of the colour distribution, and our GMM takes into account the specular component of the colour distribution. This makes our work different from many other colour modeling methods which approximate surface property as Lambertian and do not take specular reflectance into account. While a colour classifier relying on only the diffuse reflection is not capable of handling

the situation when specular reflection takes significant portion of the object surface reflection, our algorithm can maintain stability under a wide range of illumination, even when specular reflection is becoming a significant portion of the surface reflection.

Our experiment results support that a GMM of two components is an accurate representation of the colour distribution. The GMM which incorporates both diffuse and specular component of the colour distribution has better accuracy in colour classification than a colour model which only uses the diffuse component. This can be of significant importance for a general-purpose colour-vision application. For example, an outdoor computer vision application faces a more serious problem of the specular reflection of the scene caused by the outdoor illumination. An autonomous vehicle system needs to have good recognition of the road signs and other image informations, which are usually coupled with specular reflection. Using the GMM for colour modeling, it would have better performance in dealing with segmenting images with specular reflection. Another example is the human-skin tracking commonly used in a human-computer interactive system. Although a single Gaussian can be used to model the skin-colour distribution in most illumination-friendly situations, it is noted that a single Gaussian is not always sufficient [55, 65]. Using the GMM modeling with adaptive mechanism, skin-colour tracking can have a better performance.

In our algorithm we apply a simple exponentially decaying function to update the colour model using the GMM derived at each time step. Iteratively updating the GMM colour model enables our system to keep an accurate and complete representation of the colour distribution even when the illumination goes through a wide range of variation. This ensures the colour classification system to be robust under dynamic illumination. We avoid making the

linearity assumption of the colour evolution. as the linear transformation is applicable only for approximately Lambertian surfaces. For dichromatic materials the surface reflectance and the body reflectance together can not be modeled with a linear transformation.

Another advantage of our algorithm is efficiency. By reducing the dimensionality of GMM modeling from 3D to 2D, converting the GMM to probability tables in 2D, and checking the membership of each pixel within the LUTs derived from 2D probability tables, our algorithm reduces the time cost of the entire process. Currently, our system can process video feed at 30 frames/second on an off-the-shelf desktop computer. We are aware that it is possible to reduce the computational cost of evaluating a color pixel with the GMM to $O(n)$ by applying some techniques to approximate the evaluation function. For example, taking the logarithm on both sides of the Gaussian function could avoid directly evaluating the exponential function; therefore it could reduce the computational cost of evaluating a color pixel with a Gaussian function to be constant. As the probability of a color pixel in GMM is the weighted sum of Gaussian functions, directly applying logarithm can only be applied as an approximation for a GMM with well separated Gaussian components. The accuracy of this approximation decreases if there is significant overlapping among Gaussian components. The same techniques for approximation could also be applied in our algorithm to further improve efficiency. The approach taken in our algorithm is one of the methods to achieve real-time efficiency. It should be noted that our vision system converts RGB to YUV in software since the camera only provides RGB image. This conversion consumes about 15 ms per frame or about 50% of the total processing time for every frame.

The estimated GMM derived from labeled pixels during illumination changes

is only an approximation of the colour model. As the colour adaptation of our algorithm is an unsupervised process, there always exists a problem of false negative and false positive. False negatives are the pixels belonging to an interesting colour class but not being correctly classified. As lighting changes, false negatives can cause a problem for adaptation as only the labeled pixels are used as samples to measure and update the colour model; therefore the measured colour model is biased toward the previous model. Based on the observation that target colour blobs are closed regions in both image and colour space, we solve this problem with a simple region growing technique. Another issue is false positive. False positives are the noise pixels classified as an interesting colour class. In our system, a spatial constraint of the colour blobs is applied to remove false positive.

In summary, we present a study of the colour classification problem in this thesis. An adaptive colour classification algorithm is proposed, and a colour classification system for RoboCup which implements our algorithm is also presented. We use this system for fast and accurate colour classification to locate and identify the ball and all robot players on the field for RoboCup competition. The results show that our system can handle dynamic lighting condition quite reliably in the sense of efficiency and accuracy. Also our experiments show the proposed algorithm can be applied in a more general colour vision system such as human-skin colour tracking application, with a satisfying performance. Our study suggests that GMM representation of the body reflectance and the surface reflectance is a good approximation of the colour distribution of a dichromatic surface, and it can be expected to serve as an accurate model for a general colour classification application. Also the simplicity of our colour model ensures that it can be effectively adapted to dynamic lighting, such that it serves as a robust adaptive colour classification

algorithm.

Bibliography

- [1] L. Baum, T. Petrie, G. Soules, and N. Weiss. Maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Statistics*, 41(1):164–171, 1970.
- [2] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [3] C. A. Bouman. Cluster: An unsupervised algorithm for modeling Gaussian mixtures. Available from <http://www.ece.purdue.edu/~bouman>, April 1997.
- [4] D. A. Brainard and B. A. Wandell. Asymmetric color matching: how color appearance depends on the illuminant. *Journal of the Optical Society of America A*, 9 (9):1433–1448, 1992.
- [5] David H. Brainard and Brian A. Wandell. Analysis of the retinex theory of color vision. *J. Opt. Soc. Am. A*, 3(10):1651–1661, 1986.
- [6] M.H. Brill. Image segmentation by object color: a unifying framework and connection to color constancy. *J. Opt. Soc. Am. A*, 7:2041–2047, 1990.
- [7] B. Browning, M. Bowling, J. Bruce, R. Balasubramanian, and M. Veloso. Cm-dragons'01-vision-based motion tracking and heterogenous robots. In *RoboCup 2001: Robot Soccer World Cup V*, pages 567–570, London, UK, 2002. Springer-Verlag.
- [8] J. Bruce, T. Balch, and M. Veloso. Fast and inexpensive color image segmentation for interactive robots. In *IROS-2000*, 2000.
- [9] D. Comaniciu, V. Ramesh, and P. Meer. real-time tracking of non-rigid objects using mean shift. In *IEEE conference on computer vision and pattern recognition*, volume 2, pages 142–149, Jun 2000.
- [10] T. Darrell, G. G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 601–607, 1998.
- [11] M.S. Drew and L.L. Kontsevich. Closed-form attitude determination under spectrally varying illumination. In *Computer vision and pattern recognition*, pages 985–990, 1994.

- [12] A. Elgammal, R. Duraiswami, and L. Davis. Efficient nonparametric adaptive color modeling using fast gauss transform. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 2, pages 563–570, Dec 2001.
- [13] G. D. Finlayson. color in perspective. *IEEE transaction on pattern analysis and machine intelligence*, 18:1034–1038, 1996.
- [14] G. D. Finlayson, S. D. Hordley, and P. M. Hubel. color by correlation: a simple, unifying framework for color constancy. *IEEE transaction on pattern analysis and machine intelligence*, 23(11), november 2001.
- [15] G. D. Finlayson, P. M. Hubel, and S. D. Hordley. color by correlation. In *Proc. the 5th Color Imaging Conference*, pages 6–11, 1997.
- [16] D. Forsyth. A novel algorithm for color constancy. *international journal of computer vision*, 5:5–36, 1990.
- [17] G. F. Wyeth G.F. and B. Brown. *Robust Adaptive Vision for Robot Soccer*. Research Studies Press, 2000.
- [18] R. C. Gonzalez and R. E. Woods. *Digital Image processing*. Addison Wesley, 1992.
- [19] W. Hafner and O. Munkelt. Using color for detecting persons in image sequences. *Pattern Recognition and Image Analysis*, 7(1):47–52, 1997.
- [20] G. Healey. Using color for geometry-insensitive segmentation. *J. Opt. Soc. Am. A*, 6:920–937, 1989.
- [21] R. Jain, R. Kasturi, and B. G. Schunck. *Machine Vision*. McGraw-Hill, 1995.
- [22] M. Jones and J. Rehg. Statistical color models with application to skin detection, 1998. Technical Report CRG 98/11.
- [23] M. J. Jones and J. M. Rehg. Statistical color models with applications to skin detection. In *Proc. IEEE conference on computer vision and pattern recognition*, 1999.
- [24] P. Jonker, J. Caarls, and W. Bokhove. Fast and accurate robot vision for vision based motion, 2001.
- [25] H. Kitano. Research program of robocup. *Applied Artificial Intelligence*, 12(2-3), 1998.
- [26] R. Kjeldsen and J. Kender. Finding skin in color images. In *Proc. IEEE FG'96*, pages 312–317, 1996.
- [27] Gregor Klancar, Omar Orqueda, Drago Matko, and Richard Karba. robust and efficient vision system for mobile robots control-application to soccer robots. *Electrotechnical Review*, 68(5):306–312, 2001.
- [28] G. J. Klinker, S. A. Shafer, and T. Kanade. The measurement of highlights in color images. *International Journal of Computer Vision*, 2(1):7–32, 1988.

- [29] G. J. Klinker, S. A. Shafer, and T. Kanade. A physical approach to color image understanding. *International Journal of Computer Vision*, 4:7–38, 1990.
- [30] E. H. Land. the retinex theory of color vision. *scientific american*, 237:108–129, 1977.
- [31] H. Lee. Illuminant color from shading. volume 1250, SPIE, 1990.
- [32] H. Lee, E. J. Breneman, and C. P. Schulte. Modelling light reflection for computer color vision. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 4:402–409, 1990.
- [33] Scott Lenser and Manuela Veloso. Automatic detection and response to environmental change. In *Proceedings of ICRA'03, the 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, May 2003.
- [34] B. Li, H. Hu, and L. Spacek. an adaptive color segmentation algorithm for sony legged robots. In *proceedings of the 21st IASTED international conference APPLIED INFORMATICS*, Innsbruck Austria, February 10-13 2003.
- [35] Marcel Lucassen. Quantitative studies of color constancy, 1993.
- [36] L. T. Maloney. Illuminant estimation as cue combination. *Journal of Vision*, 2(6):493–504, 2002.
- [37] J. Martin, V. Devin, and J. Crowley. Active hand tracking. In *Proc. IEEE international conference on automatic face and gesture recognition*, 1998.
- [38] J. J. McCann, S. P. McKee, and T. H. Taylor. Quantitative studies in retinex theory. *vision research*, 16:445–458, 1976.
- [39] John J. McCann, Suzanne P. McKee, and Thomas H. Quantitative studies in retinex theory. *Vision Research*, 16:445–458, 1976.
- [40] S. J. McKenna, S. Jabri, Z. Duric, and A. Rosenfeld. Tracking groups of people. *Computer Vision and Image Understanding*, 80:42–56, 2000.
- [41] M. McNaughton and H. Zhang. Color vision for robocup with fast lookup tables. In *IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*, Changsha, China, October 8-14 2003.
- [42] K. Nummiaro, E. B. Koller-Meier, and L. Van Gool. Object tracking with an adaptive color-based particle filter. *Symposium for Pattern Recognition of the DAGM*, pages 355–360, 2002.
- [43] A.P. Petrov and L.L. Kontsevich. Surface colour for segmentation. *J. Opt. Soc. Am. A*, 11(10):2745–2749, 1994.
- [44] Y. Raja, S. J. Mckenna, and S. Gong. Tracking colour objects using adaptive mixture models. *Image vision computing*, 17(3-4):225–231, 1999.

- [45] C. Rasmussen, K. Toyama, and G. Hager. Tracking objects by color alone, 1996.
- [46] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM Review*, 26(2):195–239, 1984.
- [47] D. M. Rennie. Derivation of the f-measure, 2004.
- [48] J. Rissanen. A universal prior for integers and estimation by minimum description length. *annals of statistics*, 11(2):417–431, 1983.
- [49] RoboCup2004. Laws of the f180 league 2004 c release v3.00.
- [50] Tominaga S. and B. A. Wandell. Standard surface-reflectance model and illuminant estimation. *Journal of the Optical Society of America A*, 6(4):576–584, 1989.
- [51] S. A. Shafer. Using color to separate reflection components. *Color Res. App.*, 10(4):210–218, 1985.
- [52] Leonid Sigal, Stan Sclaroff, and Vassilis Athitsos. Estimation and prediction of evolving color distribution for skin segmentation under varying illumination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 152–159, 2000.
- [53] Leonid Sigal, Stan Sclaroff, and Vassilis Athitsos. Skin color-based video segmentation under time-varying illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(7):862–877, 2004.
- [54] M J. Swain and D. H. Ballard. Color indexing. *Int.J. Computer Vision*, 7(1):11–32, 1991.
- [55] J. C. Terrillon and S. Akamatsu. Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images. In *Proc. Vision Interface*, pages 180–187, 1999.
- [56] P. J. Thomas, Russel J Stonier, and P. J. Wolfs. robustness of colour detection for robot soccer. In *seventh international conference on control, automation, robotics and vision (ICARCV'02)*, Singapore, December 2002.
- [57] S. Tominaga. Realization of color constancy using the dichromatic reflection model. In *The second IS&T and SID's Color Imaging Conference*, pages 37–40, 1994.
- [58] C. J. van Rijsbergen. Information retrieval, 1979.
- [59] C. Wren, A. Azarbayejani, T. Darrel, and A. Pentland. Pfunder: Real-time tracking of the human body. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1995.
- [60] Y. Wu and T. S. Huang. Color tracking by transductive learning. In *Proc. IEEE Int'l Conf. on Comput. Vis. and Patt. Recog.*, pages 133–138, 2000.

- [61] Ying Wu and Thomas S. Huang. Nonstationary color tracking for vision-based human-computer interaction. *IEEE Transactions on Neural Networks*, 13(4), 2002.
- [62] G. Wyszecki and W.S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley, New York, 2nd edition, 1982.
- [63] Raja Y, McKenna S J, and Gong S. Colour model selection and adaptation in dynamic scenes. *European Conference of Computer Vision (ECCV)*, pages 460–474, June 1998.
- [64] J. Yang, W. Lu, and A. Waibel. Skin-color modelling and adaptation. In *Proc. of Asian Conference on Computer Vision*, pages 687–694, 1998.
- [65] M. H. Yang and N. Ahuja. Gaussian mixture model for human skin color and its application in image and video databases. In *Proc. SPIE Conf. on Storage and Retrieval for Image and Video Databases*, pages 458–466, 1999.
- [66] Qiang Zhu and Kwang-Ting Cheng. Adaptive learning of an accurate skin-color model. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG 2004)*, Korea, 2004.