Road Detection Using Intrinsic Colors in a Stereo Vision System

Dong Si Tue Cuong

Submitted in partial fulfillment of the

requirements for the degree

of Master of Engineering

in the Faculty of Engineering

NATIONAL UNIVERSITY OF SINGAPORE

2009

Acknowledgments

I would like to express my gratitude to all those who gave me the possibility to complete this thesis.

Firstly, I would like to thank my supervisor A/Prof Ong Sim Heng for his support and guidance throughout my Masters studies. A special thanks to Dr Yan Chye Hwang and DSO National Laboratories for giving me opportunity to work in this exciting robotic project, and introducing me to the world of robotics and computer vision. I would also like to thank Dr Guo Dong for his continuous feedback and guidance. Thanks to many other colleagues in robotic project team and DSO Signal Processing Lab, particularly Lim Boon Wah whose constant support and insightful comments were invaluable.

To my fellow students and colleagues who made Vision and Image Processing Laboratory such a memorable place to work, Liu Siying, Sameera Kodagoda, Teo Ching Lik, Hiew Litt Teen, Daniel Lin Wei Yan, Nguyen Tan Dat, Loke Yuan Ren, Jiang Nianjuan, and Bui Nhat Linh. In particular, thanks to Teo Ching Lik, Liu Siying, Sameera Kodagoda, Hiew Litt Teen for many insightful discussions that have expanded my little knowledge of various computer vision fields. To our laboratory technologist Francis Hoon who keeps the lab running smoothly.

Lastly, I would like to thank my family for their unconditional love and support.

Contents

Summary

This thesis describes a vision-based road extraction method for mobile robot, working in the outdoor environment with dynamic lighting changes. Most vision-based approaches to mobile robotics suffer from limitations such as limited range for stereo vision or erroneous performance against illumination changes for monocular vision. We propose a stereo visual sensor system and a long-range road extraction method that is able to accurately detect drivable road area at distances up to 50 meters, allowing more responsive and efficient path planning. The method is also adaptive to different roads, due to a self-supervised learning process: in each frame, road color samples are reliably collected from stereo-verified ground patches inside a pre-defined trapezoidal learning region. These color samples are used to construct and update the model of road color, which is a Gaussian mixture in an illumination-invariant color space. The color space is designed such that it is representative of intrinsic reflectance of the road surface, and independent of illumination source. The advantages of this approach with respect to other approaches are that it gives more robust results, extends the effective range beyond the stereo range, and, in particular, recognizes shadows on the road as drivable road surface instead of non-road areas.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

On October 26, 2007, 35 driverless cars gathered at the site of George Air Force Base to compete in the third and urban edition of the Defense Advanced Research Projects Agency (DARPA) Grand Challenge [10]. Since the DARPA Grand Challenge was started in 2004, the science and engineering communities have been greatly interested in autonomous vehicle technologies. Many advances have been achieved in the field and then have greatly increased the capabilities of autonomous vehicles.

The unmanned ground vehicle (UGV), also known as the autonomous vehicle or driverless car, is defined as a completely autonomous vehicle that can drive itself intelligently from one point to another without control or assistance from any human driver. Intelligent driving means that the vehicle has to follow the drivable path and avoid any unexpected obstacles on the road, and even has to follow traffic regulations when navigating in urban scenarios.

The history of UGV arguably started in 1977 when a vehicle built by Tsukuba Mechanical Engineering Lab in Japan drove itself and achieved speeds of up to 30 km/h by tracking white street markings. Shortly after that, in the 1980s, a vision-guided Mercedes-Benz robot van, designed by Ernst Dickmanns and his team, achieved 100 km/h on streets without traffic [12]. This huge success attracted interest from governments, and subsequently, the European Commission began funding the 800 million Euro EUREKA Prometheus Project

on autonomous vehicles (1987-1995). Meanwhile in United States, the DARPA-funded Autonomous Land Vehicle (ALV) project also achieved some similar initial successes. In 1990s, more robot vehicles were developed in both continents, and higher speed and farther driving distances had been achieved. In 1995, the Carnegie Mellon University Navlab project achieved 98.2% autonomous driving on a 5,000-km "No hands across America" trip [24]. However, robot cars in this period are semi-autonomous by nature; although achieving high-speed and much farther distances, they are still subject to sporadic human intervention, especially in difficult road situations.

In late 1990s and early 2000s, research into UGV experienced several turning points. Computers, especially portable computers, became more powerful and affordable. Several sensors and techniques, which were previously not feasible for autonomous vehicles, such as cameras and computer vision techniques, were gradually utilized. From 1996-2001, the Italian government funded the ARGO Project [38] at the University of Parma and Pavia University. The culmination of this project was a journey of 2,000 km over six days on the motorways of northern Italy, with an average speed of 90 km/h and 94% time of automatic driving. It was noted for its 54-km longest automatic stretch and the stereoscopic vision algorithms for perceiving its environment, as opposed to the popular "laser, radar" approach at that time. In 2002, the DARPA Grand Challenge competitions were announced, in which the cars are strictly required to be fully autonomous. While the first and second DARPA competitions competed over rough unpaved terrains and in a non-populated suburban setting, the third DARPA challenge, known as DARPA urban challenge, involved autonomous cars driving in an urban setting. Their million dollar prizes and international team participation have greatly energized world-wide research work into UGV technologies. In the first competition held on March 13, 2004 in the Mojave Desert region of the United States, none of the robot vehicles finished the 240 km route. Carnegie Mellon University's (CMU) Red Team travelled the farthest distance, completing 11.78 km of the course [8]. In the second competition which began on October 8, 2005 at the same venue, five vehicles successfully completed the race with Stanford University's Stanley robot crowned as the fastest vehicle. All

but one of the 23 finalists in the 2005 race surpassed the 11.78 km distance completed by the best vehicle in the 2004 race [9]. This fact illustrates tremendous advances in UGV technologies during the course of one year, largely stimulated by the Grand Challenges. Most recently, the third competition of the DARPA Grand Challenge, known as the "Urban Challenge", took place on November 3, 2007 at the site of the George Air Force Base. Out of six teams that successfully finished the entire course, CMU's entry was the fastest [10].

Figure 1.1: Stanley, the 2005 DARPA Grand Challenge winner.

1.2 Motivation

UGVs require reliable perception of its environment, especially the current road ahead, for efficient and safe navigation. Autonomous outdoor navigation is a difficult problem as the diversity and unpredictability of outdoor environments present a challenge for obstacle and road detection.

Obstacle detection and road extraction, defined as the two separate processes of detecting hazardous areas and finding the local drivable road areas, respectively, are fundamental and essential tasks for many intelligent autonomous vehicle navigation applications. Many navigation systems use obstacle detecting sensors and methods to build a traversability map that is populated with detected obstacles. Most mobile robots rely on range data for obstacle detection, such as laser range-finders (LADAR), radar, and stereo vision. Because these sensors measure the distances from obstacles to the robot, they are inherently

very relevant to the task of obstacle detection. However, none of these sensors is perfect. Stereo vision is simple but computationally expensive and sometimes could be very inaccurate. Laser range-finders and radar provide better accuracy, but are more complex and more expensive. Range sensors in general are unable to detect small or flat objects or distinguish different types of ground surfaces. They also fail to differentiate between the dirt road and adjacent flat grassy areas. In addition, range-based obstacle detection methods often have limited range. Most stereo-based methods are often unreliable beyond 12 meters [25] [30], while most LADAR-based methods have the effective range up to 20 meters [35].

Given the above limitations, especially the limited effective range, none of those navigation systems could have efficient path planning and fast navigation. Humans navigate accurately and quickly through most outdoor environments and have little problem with changing terrains and environment conditions. Apparently, humans can drive effortlessly because we are excellent in locating drivable paths, and are generally accurate for a very long range, up to 50-60 meters. Human visual performance is better, but this is not due to stereo perception, since human vision is more like a monocular imaging system at distance greater than one meter. Furthermore, humans do not need to know the exact distances to all objects on the road to effectively drive a vehicle. In most navigation scenarios, human drivers just locate distinct drivable paths with usually very few obstructing obstacles and follow along the paths consistently.

Recent research has focused on increasing the range of road detection for path planning beyond obstacle detection-based approaches. In fact, many color vision-based road extraction approaches with effective range beyond 50 meters have been proposed [7] [13] [28]. While extending effective range using range sensors would significantly increase hardware cost and system complexity, changes in vision systems are comparatively inexpensive, as camera images usually contain information far beyond the 20-meter range. In these vision-based approaches, the drivable road area is detected by classifying terrains in the far range according to color or texture of the nearby road. Although these methods extend the perception range, many of them are not robust and they usually misclassify in

the presence of shadows or complex terrain.

The primary contributions of this thesis are a stereo visual sensor system with adaptive long-range road extraction. A multiple-range architecture for perception is proposed. It combines two perception modules: long-range color-based road extraction and short-range stereo obstacle detection. The long-range module provides information about distant areas, thus enabling more efficient path planning and better speed control. Meanwhile, the short-range module provides obstacle information for obstacle avoidance.

The long-range road extraction module uses an online learning mechanism to adapt quickly to different environments. It maintains a Gaussian mixture as the basic road color model. As the vehicle moves, it keeps updating this Gaussian mixture with new color samples collected from a training region in front of the vehicle. The short-range obstacle detection is maintained to provide obstacle information, which is essential for close-range obstacle avoidance. In addition, any obstacle within the training region is detected and removed, and only ground color samples in the training region will be collected for updating the road color model.

The color-based long-range road extraction module has several novel features. Firstly, road color samples are validated as non-obstacle and non-grass before being used for color model updating. Previous methods either assume that the training area is free of obstacles [37] or use another sensor system that greatly increases system complexity [35]. Secondly, most color-based road extraction methods are not robust enough, especially in scenes with shadows, which cause parts of the road to have dissimilar colors. We propose to use an illumination-invariant color space that is representative of the intrinsic reflectance of the road surface and independent of the illumination source. By constructing and updating the color road model in this color space, the road areas can be extracted robustly, regardless of illumination changes. Shadows would not give the system a false perception of a dead-end road. Finally, a dynamic number of Gaussians are maintained to represent the road color model, depending on the driving terrain. By having a dynamic number of Gaussians, the road extraction module will give optimal and adaptive performance in different driving environments.

The long-range road extraction method has been extensively tested on numerous data sets obtained by a mobile robotic vehicle. Experiments on a robotic vehicle show that the road extraction method is able to perform robustly up to 50 meters and beyond, even with shadows on road, and perform adaptively in different driving environments.

1.3 Thesis Arrangement

In Chapter 2, we present the background material on previous works related to the central topic of this thesis. We briefly review research projects in UGVs, with the focus on vision-based perception for UGVs, in particular, previous works in color-based road detection and illumination invariant colors.

In Chapter 3, we give an overview of the visual system, our test vehicle platforms, as well as specify the output requirements. In Chapter 4, we present the short-range module which provides obstacle information and road color samples for the long-range module. In Chapter 5, the long-range module which extracts road area based on color is described. Experimental results are presented in Chapter 6.

Finally, Chapter 7 presents the contributions of this thesis, along with a discussion of possible future work.

Chapter 2

Background and Related Work

2.1 Road extraction

Many vision-based road extraction methods have been implemented during the last decades, from the project VIST in 1988 to those by DARPA's 2007 Grand Challenge participants. Therefore, the research work done on the subject of road extraction is voluminous.

In the first subsection, we will review different early color-based road extraction methods, with the focus on color information manipulation and color representation. Then, we will look into the color learning issue and its evolution to multi-range architecture for better system robustness and adaptivity.

2.1.1 Color-based approaches

Most of the approaches to extract road are based on color. One prominent research work in outdoor navigation is the Navlab projects. Navlab uses color vision as the main cue to detect the road for its road-following algorithm. In its 1988 implementation [36], the road pixels are represented by four separate Gaussian clusters. Each Gaussian cluster is characterized by a mean vector, a three-by-three covariance matrix and a *priori* likelihood number which is the expected percentage of road pixels in the contribution. Similarly, the non-road pixels are also represented by four separate Gaussian clusters. These clusters are constructed based on the color distribution of the sample road and non-road images. The confidence of a pixel with a particular color belonging to a Gaussian

cluster is computed using the Mahalanobis distance and is classified using the standard maximum-likelihood ratio test. After classification, the cluster statistics are recomputed and updated. Although the algorithm works well in various weather conditions, it however cannot deal with drastic changes in illumination between images.

In the 1993 Navlab implementation [6], the color update mechanism is improved. After a classification step similar to the 1988 version, road and off-road sample pixels are collected from fixed sample regions in image. These road and off-road sample regions are identified in the image based on the result of immediate previous classification. The sample pixels are grouped into, based on color similarity using the standard nearest mean clustering method, four road clusters for the road color model and another four clusters for the off-road color model. Each cluster is characterized by a mean vector, a covariance matrix and a number of sample pixels in the cluster. Similar to [36], the classification step is based on the maximum likelihood method. The road color model is better characterized and is updated by replacing itself with new clusters from each frame. However, since it has a long computation time and requires some overlapping between the images, the algorithm is not relevant for real-time road extraction for moderately fast vehicles.

To avoid the computation cost of clustering with 3D data, methods for dimension reduction and simpler classification have been proposed. In the VITS project [36], the authors observed that the road is predominantly brighter than the road shoulder in the blue image and darker in the red image. Subsequently, the "Red minus Blue" algorithm is proposed in which each pixel's red value is subtracted by the blue value and the resulting image is thresholded. Although the authors proposed various alternative and complementary approaches, they concluded that the "Red minus Blue" algorithm is the most dependable and used in formal demonstrations. However, it is not robust when there are abnormal color patches on the road such as dirt, tire track, and tarmac patch. Change in weather such as an overhead cloud could also cause system failure. The algorithm is apparently not adaptive to changes in the environmental conditions. Furthermore, the above observation by the authors is not always true for dif-

ferent kinds of road. Similarly, using the reduced dimension spaces, Lin *et al.* [29] proposed asphalt road segmentation in the Saturation-Intensity plane based on the observation that asphalt saturation is lower than that of the surrounding region. Such an algorithm apparently only works on asphalt road.

In another work, Chaturvedi *et al.* [4] [5] proposed road segmentation in the Hue-Saturation plane. They argued that by using the H-S space, the algorithm is able to work even with shadows as the luminance data is already removed in the Intensity data. However, it is observed that the algorithm only works well for red mud roads and with light shadows. It is not applicable for cases with strong shadows and other kinds of road.

Recently, in the DARPA Grand Challenge 2005, a self-supervised, adaptive color road extraction method was proposed [7] [35]. Similar to Navlab, the algorithm uses a Gausssian mixture model to represent road colors. However, the sampling, training and update mechanisms are greatly improved. The color samples are no longer collected from a fixed region in the image but from the projected laser road map onto the camera image. Only up to three Gaussians are used in the road color model, and there is no color model for off-road areas as off-road colors are too complex to represent. In addition, in the color update step, the previous color model is not immediately thrown away after a new color model is computed. Instead, a fixed number of Gaussians is kept in the combined color model. In each frame, the new model and the current model are compared for similarity, and Gaussians in the models are merged or discarded depending on its similarity and significance, following a well-defined update rule. The algorithm is shown to be quite adaptive, with both drastic changes such as road material and color changes or gradual changes such as illumination changes. This approach however requires data feed from laser scanners. Besides, the approach denies dealing with shadows by removing shadow areas in the image and classifying those areas as non-drivable. Although such solution is acceptable in desert environments, it is not desirable in driving environments where shadows from roadside trees are usually encountered.

2.1.2 Color learning

Most early approaches discussed in Section 2.1.1 assume that the color characteristics of drivable and obstacle regions are fixed. As a result, they cannot easily adapt to changing environments, such as [34] [36] [29] [4] [5]. Some methods are rule-based such as [29] [4] [5], while others are statistically trained off-line. In either way, those methods have to use manually labeled data to derive the rules or train the off-line models. Unfortunately, hand-labeling data requires lots of human effort, and such data limit the scope of the robot to environments where data are collected.

To overcome these limitations, self-supervised systems have been developed to reduce or eliminate the need for manually collected training data, and to improve the vision system's adaptivity to different environments. Early self-supervised systems assume that the ground immediately in front of the vehicle is traversable. The color in this known area will be learned using different statistical learning techniques. The rest of the image will then be classified to find similarly colored pixels. Early methods such as [6] [37] report encouraging successes. Most importantly, these methods show that the self-supervision paradigm not only relieves us from manual data collection and labeling but also allows the vehicle to adapt to changing environments.

The assumption that the immediate front ground is traversable in early works might be violated in many situations, especially in outdoor environments. Thus, there arises the need to verify the training area in front of the vehicle. For their winning robot entry, Stanley, in the 2005 DARPA Grand Challenge, the Stanford team proposes a multi-range architecture to solve the problem. In this architecture, multiple sensors with different coverages are used concurrently on the same vehicle. Sensors at close range are usually much more reliable as the close-range information is crucial for obstacle avoidance, while sensors at the farther range, although less reliable, usually have extended coverage as information from these sensors is usually intended for navigation planning. On Stanley, the more reliable close-range LADAR would provide learning samples to the long-range monocular camera [7] [35]. Since then, multi-range architectures have been used in various robots, not only in autonomous vehicles but also in

small mobile robots such as those in DARPA's Learning Applied to Ground Robots (LAGR) project [11].

2.2 Illumination invariance

Color plays an important role in many road detection methods. However, it is known that the colors in a scene not only depend on the reflectance properties of the objects' surfaces but also on the illumination conditions. This dependence is so strong that many color-based computer vision techniques may fail in various circumstances. Since the spectrum of the incident light upon a camera is the product of the illumination and spectral reflectance of the surface, the illumination must be removed for a stable representation of a surface's color. Humans have a remarkable ability to ignore the illumination effects when judging object appearance. We apparently have a subconscious ability to separate the illumination spectral power from surface reflectance spectral power within incoming visual signal. Many researchers have investigated this phenomenon by focusing on illumination invariant descriptions, which are features from color images that represent only the reflectance component and are relatively robust to changes of illumination conditions, i.e., illumination intensity and illumination color.

In this section, we will present background knowledge on the formation of colors in digital image and the effects of illumination colors and surface colors on the final image colors. We will also review some recent research on illumination invariant features.

2.2.1 Color formation and properties

Colors in a digital image are formed as different digitized responses of the camera system to different wavelength radiations of the incident light. In summary, the color image of an object is determined by properties of the illumination source, object's surface reflectance, image sensor, and camera system's digital coding process, as shown in Figure 2.1.

In this subsection, we will examine physical properties of colored light sources, colored surfaces and formation of color images in a digital image system.

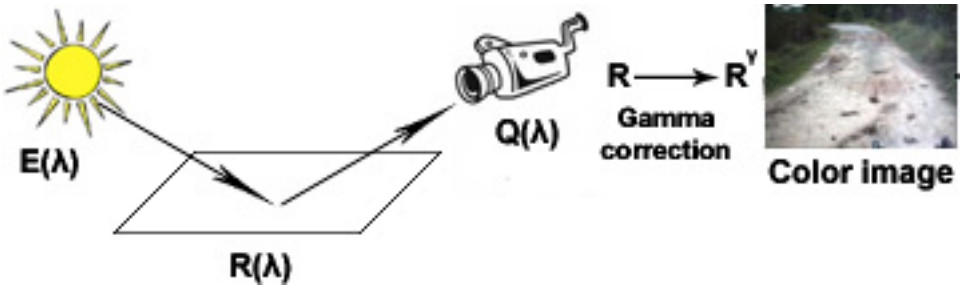


Figure 2.1: The formation of a digital color image.

2.2.1.1 Color of light sources

Light is electromagnetic radiation that is visible to human eye. Thus, as a form of electromagnetic radiation, light can be described by its wavelength and the power emitted at each wavelength. Plotting the emitted power as a function of the wavelength gives the spectral power distribution (SPD) curve of a particular light source. Common sources of light include black body radiators, the sun, the sky, and artificial illuminants.

The most basic and idealized light source is called a black body. It is an idealized object that absorbs all electromagnetic radiation that falls on it [23]. Since there is no reflected light, which is visible electromagnetic radiation, the object appears black when it is cold, and, hence, the name "black body". However, a black body emits thermal radiation when heated. On being heated, black bodies glow dull red like a hot electric stove, then become progressively brighter and whiter, like the filaments of incandescent lamps. Planck's Law states that the spectral power distribution of black body radiation depends only on the temperature of the body:

$$E(\lambda, T) \propto \lambda^{-5}(\exp \frac{hc}{kT\lambda} - 1)^{-1}, \tag{2.1}$$

where T is the temperature of the black body in Kelvin degrees, λ is the wavelength, and h, k, c are Planck's constant, Boltzmann's constant, and the light speed constant, respectively. $E(\lambda)$ represents the spectral radiance of electromagnetic radiation, which is measured in power per unit area of emitting surface per unit solid angle per unit frequency.

In the outdoor environment, the most important light source is the sun. The sun is usually modeled as a distant, bright point source. Besides the sun,

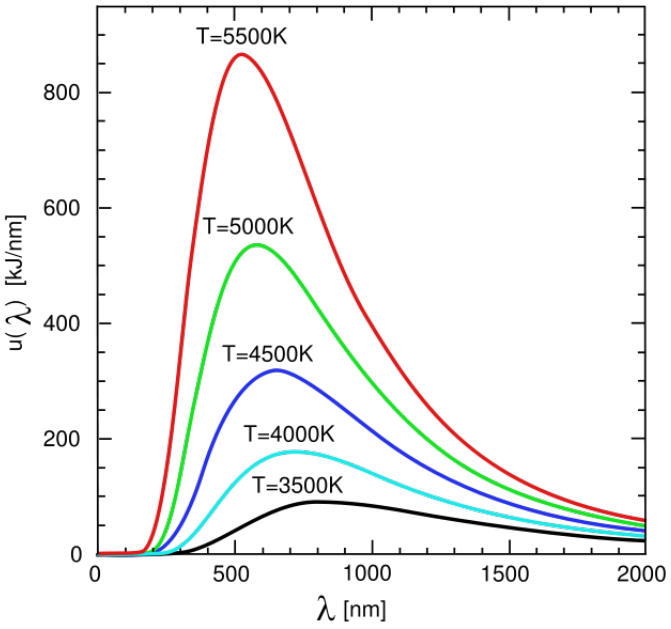


Figure 2.2: Planck's law: black body radiation spectrum.

the sky is another important natural light source. The sky is bright because sunlight from the sun is diffused upon entering the atmosphere. An outdoor surface is often illuminated by both direct sunlight from the sun and diffused light from the sky. Although these natural light sources are not black body radiators, they can be represented as a virtual black body with a determined temperature, called *correlated color temperature* or *color temperature*. It is determined by comparing the light sources' chromaticity with that of an ideal black body radiator. The temperature at which the heated black body matches the color of the light source is the light source's color temperature. Based on this definition, a number of spectral power distributions have been defined by the International Commission on Illumination (CIE) for use in describing color [41]. These distributions are known as standard illuminants [42]. For example, incandescent light is represented by the standard illuminant A, equivalent to a black body radiator with a color temperature of approximately 2856 K. In our case, natural daylight is defined as standard illuminants D, replacing deprecated B and C illuminants to simulate daylight. In fact, D65 standard illuminant, with a color temperature of approximately 6500 K as shown in Figure 2.3, is the most commonly adopted in industries to represent daylight [40].

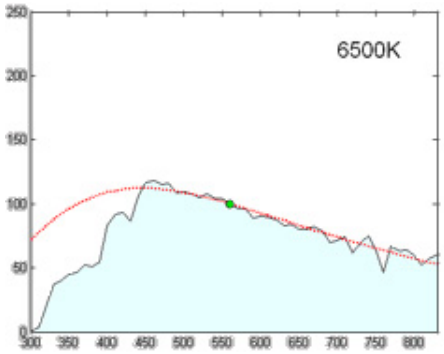


Figure 2.3: Relative SPD of D65 illuminant (black) and a black body of color temperature 6500 K (red). Retrieved from [40].

2.2.1.2 Color of surfaces - Reflectance

The color of a surface is determined by the absorption and reflection properties of the surface to different wavelength light radiation. The process is inherently complex but it is usually simplified and modeled by a bidirectional reflectance distribution function (BRDF). BRDF is a 4-dimensional function that defines how light is reflected at an opaque surface, usually as the ratio of spectral radiance in the outgoing direction to the spectral irradiance in the incoming direction.

For an outdoor road surface, we are interested in the Lambertian surface model, in which the BRDF is a constant. The reflected radiance from the surface is independent of outgoing direction. That means the apparent brightness of a Lambertian surface to an observer is the same, regardless of the observer's angle of view. The Lambertian surface model represents a perfectly diffuse surface, and it is a good approximation of any rough surface such as a dry road surface.

In contrast with Lambertian surface, a specular surface only has reflected radiance leave along a specular direction. The specular surface model represents a mirror or a glossy surface. An ideal specular surface behaves like an ideal mirror; if the viewer is not in the specular direction, the reflected specular light will not be seen. For outdoor roads, specular surfaces can be encountered as water puddles or wet tarmac road surfaces. In our project, water puddles are defined as not drivable, and wet tarmac road sections are rarely encountered. Therefore, we can safely assume that road surfaces are composed of local Lambertian patches.

2.2.1.3 Formation of color image - Sensor output

The image of an object is formed as light radiation reflected from its object surface enters an imaging system. From the above discussion, it is clear that the reflected light is determined by two factors: the light source's spectral power distribution and the surface's spectral reflectance. In addition, for a digital imaging system, the colors of an object in the final digital image is also determined by the digitized responses of the image sensors in the camera to the incident light. The sensor's output signal strength depends not only on the intensity of the incoming light signal but also on the wavelength components of the incoming light signal. Plotting the ratio of the output power to the input power as a function of the wavelength gives the spectral response curve of an image sensor.

For digital color cameras, especially the high-quality models, there are generally three image sensor components, corresponding to the red, green, and blue channels. Each image sensor is designed to respond more strongly to a particular range of color, and thus, they have different spectral responses. Figure 2.4 shows spectral responses of image sensors in the Bumblebee2 camera used in our project.

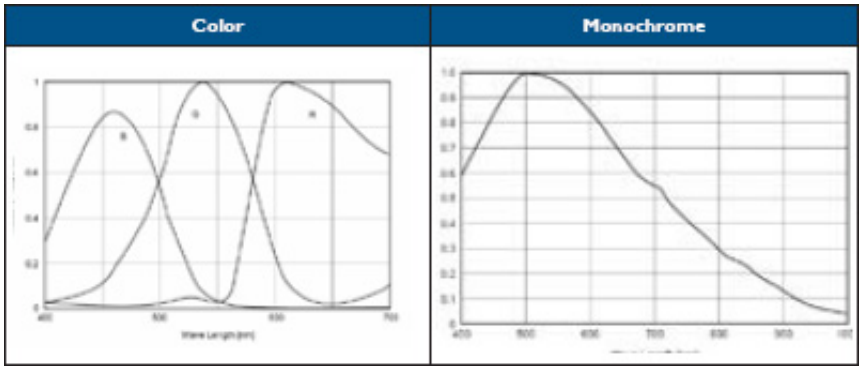


Figure 2.4: Spectral responses of Bumblebee2's sensors. Retrieved from [30].

There are several mathematical models that have been proposed for the sensor response. The most common model is the *linear response model*. In this model, it is assumed that the image sensor responses are *linear* with respect to source intensity. This *response linearity assumption* means that we could use a single spectral sensitivity function, or spectral response function, to characterize how the camera responds to sources with different spectral power distribu-

tions. Nowadays, the image sensors in most modern digital cameras are based on charge-coupled device (CCD) or active pixel sensor (APS, also known as CMOS) technology. These devices are known to have linear intensity response function over a wide operating range [39], and thus the response linearity assumption is plausible.

In the linear response model, the camera response at a pixel of an image sensor is described by an integral over the sensor response spectrum:

$$\Phi = e \int_{\lambda_l}^{\lambda_h} I(\lambda) Q(\lambda) d\lambda + n, \tag{2.2}$$

where $Q(\lambda)$ is the spectral sensitivity function of the sensor, $I(\lambda)$ is the spectral power distribution of the incident light at that particular pixel describing the power density per unit time at wavelength λ, e is the exposure duration, and n represents noise signal. λ_l and λ_h are lower and upper bound of the sensor response spectrum, respectively. It should be noted that the sensor response spectrum is possibly beyond the visible spectrum, such as those in infra-red cameras.

For our Bumblebbe2 camera and outdoor illumination, we assume that the noise is relatively minimal. In addition, as mentioned above, there are typically three sensors (red, green, blue or R, G, B) in color cameras. Thus, we have:

$$\Phi_k = e \int_{\lambda_l}^{\lambda_h} I(\lambda) Q_k(\lambda) d\lambda,\ k = R, G, B. \tag{2.3}$$

As discussed above, the reflected light, and thus $I(\lambda)$, is determined by two factors: the light source and the surface reflectance. If a perfect Lambertian surface with spectral reflectance $S(\lambda)$ is illuminated by a light source with spectral power distribution of $E(\lambda)$, the spectral power distribution of the reflected light is defined as:

$$I(\lambda) = \sigma S(\lambda) E(\lambda) \tag{2.4}$$

where σ is the shading term which is dependent only on illumination direction. In the outdoor environment, as the illumination sources are the sun and the sky, we can safely assume that σ is constant for the road surface area. Thus, after plugging Equation (2.4) into Equation (2.3) and moving the constant σ out of the integral, the camera response for an outdoor road surface is:

$$\Phi_k = \sigma e \int_{\lambda_l}^{\lambda_h} E(\lambda) S(\lambda) Q_k(\lambda) d\lambda,\ k = R, G, B. \tag{2.5}$$

The constant σe in the above equation will be ignored, as we are only interested in the relative strength of the camera response. From Equation (2.5), it is apparent that illumination changes such as shading, shadows, and specularities as well as local surface reflectance variation will introduce changes in the apparent road color in the image. This makes the road segmentation and navigation task in outdoor environments more difficult.

2.2.1.4 Formation of color image - System output

As previously discussed, an image taken by a digital color camera will have its color, or sensor responses, described by:

$$\Phi_k = \int_{\lambda_l}^{\lambda_h} E(\lambda)S(\lambda)Q_k(\lambda)d\lambda,\ k = R, G, B. \tag{2.6}$$

Suppose that the image sensor's spectral responses are very narrow-band such that they can be approximated by a Dirac delta function $Q_k(\lambda) = q_k\delta(\lambda - \lambda_k)$, where q_k represents the sensor strength, as shown in Figure 2.5. Experiments show that this approximation works well for various camera systems, especially good-quality camera systems.

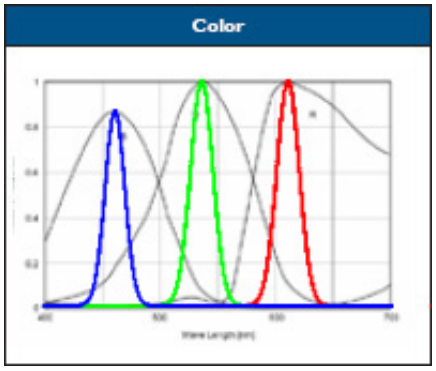


Figure 2.5: Spectral responses and their approximations by Dirac delta functions.

Using the Dirac delta function approximation, Equation (2.6) will be simplified to:

$$\Phi_k = q_k E(\lambda_k)S(\lambda_k),\ k = R, G, B, . \tag{2.7}$$

Equation (2.7) shows that the pixel values are assumed to have a linear relationship with the light source's intensity. This agrees with the sensor response linearity assumption, presented in Subsection 2.2.1.3. However, while image

sensors have a linear response, the *overall camera system*'s response may not necessarily exhibit linearity. There may be a non-linear mapping between the raw image sensor output and the final digital responses actually presentable on the camera. The most common such non-linear process is *gamma correction*. Gamma correction is a nonlinear operation used to code and decode luminance, commonly found in video or still image systems. In the simplest cases, gamma correction is defined by the following expression:

$$\Phi_{out} = \Phi_{in}^{\Gamma} \tag{2.8}$$

where Γ is known as the gamma value. A gamma value $\Gamma < 1$ is called an encoding gamma; and conversely, a gamma value $\Gamma > 1$ is called a decoding gamma. Non-linear operations such as gamma correction are designed into a camera system as the dynamic response range of the sensor is usually larger than the digital encoding range of the camera. As part of the camera digital coding process, the gamma value is changing and dependent on the overall device system as well as the individually captured image.

2.2.1.5 Color change equation

Changes in illumination color and intensity will lead to changes in sensor output, and thus, gamma value. From Equations (2.7) and (2.8), for each sensor response, i.e. color triplets (R_i,G_i,B_i), after illumination changes, the new sensor responses ($R_i^{'}$,$G_i^{'}$,$B_i^{'}$) would be:

$$\begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} \rightarrow \begin{pmatrix} R_i^{'} \\ G_i^{'} \\ B_i^{'} \end{pmatrix} = \begin{pmatrix} a^{\gamma} R_i^{\gamma} \\ b^{\gamma} G_i^{\gamma} \\ c^{\gamma} B_i^{\gamma} \end{pmatrix} = \begin{pmatrix} a' R_i^{\gamma} \\ b' G_i^{\gamma} \\ c' B_i^{\gamma} \end{pmatrix} \tag{2.9}$$

where $a' = a^{\gamma}$, $b' = b^{\gamma}$, $c' = c^{\gamma}$. γ is change ratio of gamma values Γ, and a, b, c are change ratios of image sensor outputs as illumination changes. As the sensor's spectral responses are different, changes in illumination color may cause different changes in the outputs of different sensors. Therefore, a, b, c are generally different and independent in that case, i.e. $a \neq b \neq c$. Meanwhile, changes in illumination intensity usually cause proportional changes in the sensor outputs, i.e. $a = b = c$.

Equation (2.9) reflects how RGB color values from the same surface change with changes in illumination intensity and gamma. In the following sections, this equation will be used to analyze the efficiency of the proposed illumination-invariant features.

2.2.2 Related works in illumination-invariance

In this sub-section, we will review prior research in illumination-invariance which attempts to separate surface reflectance information $S(\lambda)$ from illumination information $E(\lambda)$ given pixel color information Φ_k (as in Equation (2.6)).

2.2.2.1 General illumination-invariance research works

The importance of being able to separate illumination effects from reflectance has been well understood for a long time. Barrow and Tenenbaum [2] introduced the notion of "finding intrinsic images" to refer to the process of decomposing an image into two separate images, one image containing variation in surface reflectance and another representing the variation in the illumination across the image (or shading). In their paper [2], they proposed methods for deriving such intrinsic images under certain simple models of image formation. However, the complex nature of image formation means that such a method of recovering intrinsic images has become invalid. Later algorithms, such as the Retinex and Lightness algorithms by Land [27], were also based on other simple assumptions, such as the assumption that the gradients along reflectance changes have much larger magnitudes than those caused by shading. That assumption may be invalid in many real images, so more complex methods have been proposed to separate shading and reflectance [26] [33].

Although work on intrinsic images has attracted much attention, several computer vision applications do not need both intrinsic images. In fact, in many vision applications, it is more attractive to simply estimate and remove the effects of the prevalent illuminant in the scene rather than obtain separate surface reflectance and illumination shading information. Among various approaches to this problem is the color constancy approach. To remove the effects of illumination from the image, invariant quantities are derived from image values such

that those quantities remain unchanged under different illumination conditions. Thus, compared to conventional intrinsic image methods such as in [2] [33], this approach would effectively give only a single intrinsic image, instead of two, that contains surface reflectance information. This intrinsic image proves to be useful enough to many computer vision applications, especially in color-based image segmentation.

There are different ways of devising invariant features. A common direction is to normalize each image pixel to some reference RGB such that the new color values are invariant to lighting changes. In these methods, illumination change is often represented as a scaling factor, and it would be cancelled out in the normalized color values. In other methods such as [22], some global statistical features of the color distribution in the image are proposed to be independent of illumination. In this survey, we only look into the most prominent illumination-invariant features that have been proposed and frequently used in lighting-invariant applications. They are: normalized RGB [20], Hue in HSI or HSV color space [4], brightness-invariant features by Ghurchian [20], gray-world normalization [18], MaxRGB normalization [26], Log Hue [17], and intrinsic color [16].

In the next section, we will present computational formula for each feature and briefly analyze its effectiveness in illumination invariance, based on Equation (2.9). It appears that most common supposedly illumination-invariant features are not really invariant to illumination, and many of them do not account for changes in the gamma correction process.

Normalized RGB The normalized RGB color space is defined by:

$$(r, g, b) = (\frac{R}{R+G+B}, \frac{G}{R+G+B}, \frac{B}{R+G+B}) \tag{2.10}$$

By using Equation (2.9), it can be seen that this color space is not illumination-invariant. For each triplet (R, G, B) and corresponding normalized values (r, g, b), the new triplet (R', G', B') when illumination changes (defined by Equation (2.9)) will yield the new (r', g', b') that are not the same as (r, g, b). Only when gamma value γ and illumination color do not change, the normalized RGB becomes invariant to changes in illumination intensity, or brightness.

Normalized RGB has been known for removing effects of brightness and shading, the latter of which is dependent on the incoming direction of the illumination source. However, in outdoor environments, as the main light sources are the sun and the sky which have relatively constant illumination direction, this color space would not have a significant effect.

Hue in HSV, HSI color space HSV and HSI color space are popular color spaces. Hue is well defined by [4]:

$$H = \tan^{-1}(\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)}) \tag{2.11}$$

HSI and HSV color spaces are designed to describe perceptual color relationship more accurately than RGB color space. Hue is often used as an illumination-invariant feature as it is expected to be separated from illumination information. However, similar to normalized RGB color space, the Hue color is only brightness-invariant, and not fully illumination-invariant.

Brightness-invariant features In his paper [20], Ghurchian et. al. proposed the following "brightness invariant color parameters":

$$(r'_1, r'_2, r'_3) = (\frac{\max(G,B)-R}{\max(R,G,B)}, \frac{\max(R,B)-G}{\max(R,G,B)}, \frac{\max(R,G)-B}{\max(R,G,B)}) \tag{2.12}$$

where $\max(a,b,c)$ gives the largest value among the input values. Ghurchian et. al.'s work deals with autonomous navigation of a mobile robot in forest roads where shadows and highlights are frequently found on the road. It is claimed in the paper that these features sometimes yield better segmentation in forest road scenes than other conventional features such as normalized RGB or Hue color. However, from Equation (2.9), we can see that although those features are brightness-invariant, they are not fully illumination-invariant.

Gray-world normalization According to [44], the gray-world normalization is defined by:

$$(r^{new}, g^{new}, b^{new}) = (\frac{R}{\text{mean}(R)}, \frac{G}{\text{mean}(G)}, \frac{B}{\text{mean}(B)}) \tag{2.13}$$

where $\text{mean}(R)$, $\text{mean}(G)$, and $\text{mean}(B)$ are the average values of all red, green, blue pixels, respectively. Inserting into Equation (2.9), it is clear that no matter

how illumination color or intensity changes, the scaling factors a', b', c' will be cancelled out. However, changes in gamma correction are not considered and dealt with. Therefore, gray-world normalization is only effective for small changes in illumination color or intensity. When illumination changes are large such that gamma value changes significantly, the gray-world normalization is no longer illumination-invariant.

Max RGB normalization In the Retinex algorithm [26], an image can be normalized by dividing each color of every pixel by the largest values of that color in the whole image. This algorithm is expressed by:

$$(r^{new}, g^{new}, b^{new}) = (\frac{R}{\max(R)}, \frac{G}{\max(G)}, \frac{B}{\max(B)}) \tag{2.14}$$

where $\max(R)$, $\max(G)$, and $\max(B)$ are the largest red, green, blue color values in the image. Similar to gray-world normalization, when applied to Equation (2.9), it is clear that such normalization is only effective for small changes in illumination color and intensity.

Log Hue Given the limitations of Hue color as discussed above, Finlayson et. al. [17] proposed a variant of Hue color, called Log Hue, defined by:

$$H = \tan^{-1}(\frac{\log R - \log G}{\log R + \log G - 2\log B}) \tag{2.15}$$

Compared to the conventional Hue formula, the Log Hue color is designed to be invariant to both brightness and gamma. Indeed, by plugging this formula into Equation (2.9), we see that Log Hue color is nearly unchanged as illumination changes:

$$\text{As} \begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} \rightarrow \begin{pmatrix} a^\gamma R_i^\gamma \\ b^\gamma G_i^\gamma \\ c^\gamma B_i^\gamma \end{pmatrix}$$

$$H_i = \tan^{-1} \frac{\log R_i - \log G_i}{\log R_i + \log G_i - 2\log B_i} \rightarrow H_i' = \tan^{-1} \frac{\log(a^\gamma R_i^\gamma) - \log(b^\gamma G_i^\gamma)}{\log(a^\gamma R_i^\gamma) + \log(b^\gamma G_i^\gamma) - 2\log(c^\gamma B_i^\gamma)}$$

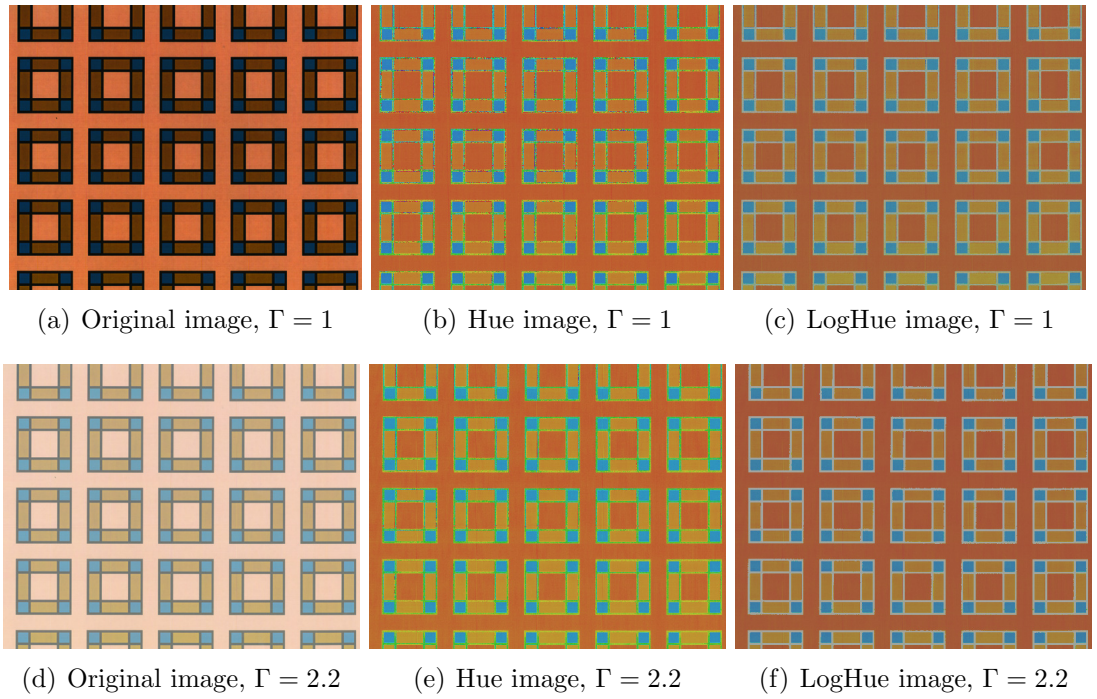
$$\text{Thus, } H_i' = \tan^{-1}(\frac{\gamma(\log R_i - \log G_i) + \gamma(\log a - \log b)}{\gamma(\log R_i + \log G_i - 2\log B_i) + \gamma(\log a + \log b - 2\log c)})$$

$$\text{Simplified, } H_i' = \tan^{-1}\left(\frac{(\log R_i - \log G_i) + (\log a - \log b)}{(\log R_i + \log G_i - 2\log B_i) + (\log a + \log b - 2\log c)}\right) \tag{2.16}$$

When $(\log a - \log b) \ll (\log R_i - \log G_i)$ and $(\log a + \log b - 2\log c) \ll (\log R_i + \log G_i - 2\log B_i)$:

$$\Rightarrow H_i' \simeq \tan^{-1}\frac{\log R_i - \log G_i}{\log R_i + \log G_i - 2\log B_i} = H_i \tag{2.17}$$

We can see that gamma factor γ is cancelled out. Thus, the Log Hue color is invariant to gamma correction. In addition, when brightness, i.e. illumination intensity, changes, the scaling factors a, b, c are identical, and H_i' is exactly equal to H_i. Thus, Log Hue color is indeed invariant to brightness and gamma, as claimed by the authors and illustrated in Figure 2.6. However, when illumination color changes significantly, the scaling factors a', b', c' may not be equal and Equation (2.17) may no longer hold. Therefore, Log Hue color is not completely illumination-invariant and would be inadequate for our outdoor applications.



(a) Original image, $\Gamma = 1$ (b) Hue image, $\Gamma = 1$ (c) LogHue image, $\Gamma = 1$

(d) Original image, $\Gamma = 2.2$ (e) Hue image, $\Gamma = 2.2$ (f) LogHue image, $\Gamma = 2.2$

Figure 2.6: Invariance comparison of Hue and Log Hue. Retrieved from [17]. The images 2.6(c) and 2.6(f) look much closer to each other than 2.6(b) and 2.6(e).

Intrinsic color In his paper [16], Finlayson proposed an invariant feature, called *reflectance intrinsic color* or *intrinsic color*, which attempts to separate illumination and reflectance components in Equation (2.6). The final output represents the intrinsic reflectance of the surface and, thus, it is fully invariant to illumination. In this method, from each triplet of sensor responses at a pixel, corresponding to red, green, blue values, the invariant feature is computed as:

$$\zeta = \log(R/G)\cos\theta + \log(B/G)\sin\theta. \tag{2.18}$$

The method is based on the assumptions of Lambertian surface, illuminants following Planck's law, and narrow-band camera sensor spectral responses following Dirac's delta function. A crucial parameter of this method is the angle of invariance θ. Originally, this angle was obtained via a calibration procedure, involving using the calibrated camera to capture images in different illumination conditions. Subsequently, it was shown [15] that the angle can be retrieved through an automatic process based on the observation that the projection in the correct θ angle will minimize the entropy in the resulting invariant image.

By applying Equation (2.18) to Equation (2.9), we see how intrinsic color changes as illumination changes:

$$\text{As} \begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} \rightarrow \begin{pmatrix} a^\gamma R_i^\gamma \\ b^\gamma G_i^\gamma \\ c^\gamma B_i^\gamma \end{pmatrix}$$

$$\zeta = \log(R/G)\cos\theta + \log(B/G)\sin\theta \rightarrow \zeta' = \log(\frac{a^\gamma R_i^\gamma}{b^\gamma G_i^\gamma})\cos\theta + \log(\frac{c^\gamma B_i^\gamma}{b^\gamma G_i^\gamma})\sin\theta$$

$$\begin{aligned} \Rightarrow \zeta' &= \gamma(\log(\frac{aR_i}{bG_i})\cos\theta + \log(\frac{cB_i}{bG_i})\sin\theta) \\ &= \gamma((\log\frac{a}{b} + \log\frac{R}{G})\cos\theta + (\log\frac{c}{b} + \log\frac{B}{G})\sin\theta) \\ &= \gamma(\log\frac{R}{G}\cos\theta + \log\frac{B}{G}\sin\theta) + \gamma(\log\frac{a}{b}\cos\theta + \log\frac{c}{b}\sin\theta) \\ &= \gamma\zeta + 0 = \gamma\zeta \end{aligned}$$

as θ is retrieved such that $\log\frac{a}{b}\cos\theta + \log\frac{c}{b}\sin\theta = 0$.

So, as illumination changes, the intrinsic value varies proportionally by gamma value, independent of illumination. This result is significant as usually the gamma value Γ changes slowly and the ratio γ is quite close to 1. Furthermore, for intra-image illumination changes such as shadows, the gamma value Γ is unchanged, and $\gamma = 1$. For applications such as color-based classification, such linear variation can be overcome by normalizing the image. Thus, the intrinsic color is invariant to illumination and nearly invariant to gamma correction.

Although real light will not completely follow Planck's law, nor will the camera sensor's spectral response be narrow like the Dirac's delta function, the method works well as these assumptions are approximately true for outdoor scenes and most good-quality or high-end camera systems. This intrinsic color proves to be robust enough, especially for high-end camera systems, and it has been used in various shadow-removal applications.

2.2.2.2 Summary of invariant features and application to shadows

Tables 2.1 and 2.2 summarize the illumination-invariant features and their invariance properties. Most invariant features are designed to predict changes by illumination and try to compensate for such changes. However, these approaches only focus on changes in illumination intensity, or brightness, and fail to consider changes in illumination color. In fact, most illumination-invariant features are derived by assuming that there is only a single illuminant or equivalently multiple similar light sources concurrently illuminating. Thus, effectively the overall illumination color is fairly similar while only illumination intensity is changing. In practice, especially for outdoor environments, that is not the case. There are typically two light sources in the outdoor scene: sunlight and skylight. In outdoor environments, while non-shadow regions are illuminated by both sunlight and skylight, the shadow regions are illuminated by skylight only. As the sun and the sky have different color temperatures (Subsection 2.2.1.3), their illumination colors are generally different. Thus, between shadow and non-shadow regions, not only illumination intensity but also illumination color is different. In the case of illumination color change, features such as normalized RGB, Hue or Log Hue may be not invariant, and, thus, they are not shadow-invariant, as discussed

above.

Meanwhile, some invariant features use global statistics retrieved from the whole image as a scaling factor. For example, Gray-world normalized and Max RGB normalized colors use mean and max pixels values, respectively, as their common divisor. While these methods are able to remove effects from illumination, although not from gamma correction, they are only effective for inter-image illumination changes. For illumination changes within a single image, such as shadows, such methods have no effect as shadow and non-shadow colors after scaling by a common factor are still significantly different.

In contrast with previous invariant features, the intrinsic color feature attempts to separate illumination and reflectance components in the reflected light. The final obtained value represents the intrinsic reflectance of the surface, and thus, it is closest to shadow-invariance, as shown in Table 2.2. Therefore, we adopt the intrinsic color space in our robotic application.

2.2.2.3 Illumination-invariance in outdoor robotics

While color-based road extraction methods work well most of the time, as discussed in Section 2.1.1, they are not the complete solutions to outdoor road extraction problem. Among the main hazards to color-based road classification are shadows on the road. The road classification is based on the hypothesis that road color is similar in the whole scene. Since the shadows have very different colors from the rest of the road, it is often misclassified as non-road. Such behavior is not acceptable for navigation in outdoor environments where shadows are frequently encountered such as jungle tracks or urban roads.

Several color-based road detection methods have been proposed to be invariant to shadows for outdoor mobile robots. Based on the observation that shadows significantly change the brightness of an area without significantly modifying the color information, those methods exploited computational color measures that separate the brightness from the chromatic components. Various works in general illumination-invariance research such as "intrinsic image" works as well as illumination-invariant features have been applied with different degrees of success. The common approaches are to perform road segmentation in another color

space rather than RBG color space, such as HSV, HLS, and L*a*b [4] [5]. In these color spaces, it is believed that brightness information is represented in an Intensity/Brightness channel and chromatic information is represented in other channels. Thus, the image is converted from RGB color space into these color spaces. Then, color learning and classification is performed on chromatic color channels. Hue is often used as the illumination-invariant feature in these cases as it is expected to be unchanged between shadow and illuminated regions. However, experiments show that such approaches work only in small variances of brightness such as in Figure 2.7(b); they perform poorly with dark shadows such as in Figure 2.7(a). In particular, Hue as an illumination-invariant feature was proposed in [4] [5]. When experimenting on real outdoor data, the Hue value is generally unstable and unreliable at the very high or low brightness value, leading to erroneous segmentation with many false positives. Other research works also mentioned similar observations, such as in [20]. This could be explained by the fact that changes in gamma correction and illumination color were not considered and discounted (Section 2.2.2.1). Similarly, in another work by Ghurchian [20], the proposed brightness-invariant features also failed to discount changes in gamma correction and illumination color. Therefore, although those features are claimed to give better results than conventional features such as normalized RGB, they are not robust enough.

(a) Dark shadow (b) Light shadow

Figure 2.7: Difference between dark shadow and light shadow.

In an earlier approach [14], we proposed that RGB color space could still be used for road color learning and classification, in contrast with [4] [20]. How-

ever, during the color learning step, we tried to detect RGB color samples that are associated with shadows on road by using Log Hue color [17]. We observed that in an RGB-color-based road extraction method [7], RGB color information of shadows are usually collected but discarded after a few frames since the shadow models usually have much fewer color samples. By using a dynamic number of color models and detecting those models corresponding to shadow's RGB colors, we classify the shady roads in RGB color space. Although the method provides acceptable outputs against shady roads, it is however not efficient enough, for a number of reasons. Firstly, Log Hue color is not a highly illumination-invariant feature, as discussed in Section 2.2.2.1. Therefore, there is chance, although small, that the RGB color model for shadows is incorrectly constructed. Secondly, the RGB color model for shadows must be constructed before the shadows can be correctly classified. Thus, color samples for the shadows must be collected beforehand. Furthermore, if shadows are rarely encountered on the road, it is possible that the shadow color model will be gradually become obsolete and discarded. Then, any new shadows on the road will be misclassified until color samples of shadows are collected. During that time, the vehicle has to rely on another sensor such as stereo module as proposed in this same method [14] to navigate and collect shadow color samples, which is slower and undesirable. Finally, in the classification stage, as an extra RGB color model is kept for shadows, any color pixel would be generally verified against both color models for road and shadow. As a result, the method is much more computationally expensive.

From the above discussion, it is clearly desirable for us to perform road classification in a *truly* illumination-invariant color space. In that case, we need to maintain and update only one color model that represents road surface reflectance. With single color model, classification will become more computationally efficient. In addition, any collected road samples can be used to update this model. We also do not have to learn shadow colors beforehand and update them separately.

Table 2.1: Comparison of illumination-invariant features

Name of invariant features	Description	Color change equation $\begin{pmatrix} R_i \\ G_i \\ B_i \end{pmatrix} \rightarrow \begin{pmatrix} a^\gamma R_i^\gamma \\ b^\gamma G_i^\gamma \\ c^\gamma B_i^\gamma \end{pmatrix} = \begin{pmatrix} a' R_i^\gamma \\ b' G_i^\gamma \\ c' B_i^\gamma \end{pmatrix}$
Normalized RGB [20]	$(r,g,b) = (\frac{R}{R+G+B}, \frac{G}{R+G+B}, \frac{B}{R+G+B})$	$(r',g',b') = (\frac{a'R^\gamma}{a'R^\gamma+b'G^\gamma+c'B^\gamma}, \frac{b'G^\gamma}{a'R^\gamma+b'G^\gamma+c'B^\gamma}, \frac{c'B^\gamma}{a'R^\gamma+b'G^\gamma+c'B^\gamma})$ When $\gamma = 1$, $a' = b' = c'$, $(r',g',b') = (\frac{R}{R+G+B}, \frac{G}{R+G+B}, \frac{B}{R+G+B}) = (r,g,b)$
Hue color [4]	$H = \tan^{-1}(\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)})$	$H' = \tan^{-1}(\frac{\sqrt{3}(b'G^\gamma - c'B^\gamma)}{(a'R^\gamma - b'G^\gamma)+(a'R^\gamma - c'B^\gamma)})$ When $\gamma = 1$, $a' = b' = c'$, $H' = \tan^{-1}(\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)}) = H$
Brightness-invariant feature [20]	$(r_1,r_2,r_3) = (\frac{\max(G,B)-R}{\max(R,G,B)}, \frac{\max(R,B)-G}{\max(R,G,B)}, \frac{\max(R,G)-B}{\max(R,G,B)})$	$(r_1',r_2',r_3') = (\frac{\max(b'G^\gamma, c'B^\gamma)-a'R^\gamma}{\max(a'R^\gamma, b'G^\gamma, c'B^\gamma)}, \frac{\max(a'R^\gamma, c'B^\gamma)-b'G^\gamma}{\max(a'R^\gamma, b'G^\gamma, c'B^\gamma)}, \ldots)$ When $\gamma = 1$, $a' = b' = c'$, $(r_1',r_2',r_3') = (\frac{\max(G,B)-R}{\max(R,G,B)}, \frac{\max(R,B)-G}{\max(R,G,B)}, \frac{\max(R,G)-B}{\max(R,G,B)}) = (r_1,r_2,r_3)$
Gray-world normalization [18]	$(r,g,b) = (\frac{R}{\text{mean}(R)}, \frac{G}{\text{mean}(G)}, \frac{B}{\text{mean}(B)})$	$(r',g',b') = (\frac{R^\gamma}{\text{mean}(R^\gamma)}, \frac{G^\gamma}{\text{mean}(G^\gamma)}, \frac{B^\gamma}{\text{mean}(B^\gamma)})$
Max RGB normalization [26]	$(r,g,b) = (\frac{R}{\max(R)}, \frac{G}{\max(G)}, \frac{B}{\max(B)})$	$(r',g',b') = (\frac{R^\gamma}{\max(R^\gamma)}, \frac{G^\gamma}{\max(G^\gamma)}, \frac{B^\gamma}{\max(B^\gamma)})$
Log Hue [17]	$H = \tan^{-1}(\frac{\log R - \log G}{\log R + \log G - 2\log B})$	$H' = \tan^{-1}(\frac{(\log R - \log G)+(\log a - \log b)}{(\log R + \log G - 2\log B)+(\log a + \log b - 2\log c)})$ When $a' = b' = c'$, $H' = \tan^{-1}\frac{\log R - \log G}{\log R + \log G - 2\log B} = H$
Intrinsic color [16]	$\zeta = \log(R/G)\cos\theta + \log(B/G)\sin\theta$	$\zeta' = \gamma(\log\frac{R}{G}\cos\theta + \log\frac{B}{G}\sin\theta) + \gamma(\underbrace{\log\frac{a}{b}\cos\theta + \log\frac{c}{b}\sin\theta}_{=0}) = \gamma\zeta$

Table 2.2: Comparison of illumination-invariant features (cont.)

Name of invariant features	Invariance to illumination intensity $a' = b' = c'$	Invariance to illumination color $a' \neq b' \neq c'$	Invariance to gamma correction $\gamma \neq 1$	Remarks
Normalized RGB [20]	Yes, when $\gamma = 1$	No	No	Invariant to brightness
Hue color [4]	Yes, when $\gamma = 1$	No	No	Invariant to brightness
Brightness-invariant feature [20]	Yes, when $\gamma = 1$	No	No	Invariant to brightness
Gray-world normalization [18]	Yes	Yes	No	Not invariant to intra-image changes, e.g. shadows
Max RGB normalization [26]	Yes	Yes	No	Not invariant to intra-image changes, e.g. shadows
Log Hue [17]	Yes	No	Yes	Invariant to brightness and gamma
Intrinsic color [16]	Yes	Yes	Yes, when linearly normalized	Invariant to illumination source and gamma

Chapter 3

System Overview

3.1 The robot platform

The vision system described here was developed and mounted on a Polaris's Ranger vehicle platform, as shown in Figure 3.1. The vehicle is well-suited for off-road conditions and has a maximum speed of 30 km/h. Also mounted on the vehicle are processing units which are on-board computers, running in Linux operating system.

Figure 3.1: The vehicle platform.

For visual sensor, the used sensor is a Bumblebee2 camera (Figure 3.2). The detailed specifications of the camera are found in [30]. The camera was chosen for its stability, good image quality and support in both Windows and

Linux environment. The cameras come pre-calibrated and the Software Development Kit (SDK) supplied by the manufacturer comes with stereo processing algorithms and image rectification functions.

Figure 3.2: Bumblebee2 stereo camera sensor.

As the Bumblebee2 camera is a qualified IEEE-1394 compliant product, the libraries libdc1394 and libraw1394 are necessary to control the FireWire bus to capture the images in Linux. These operations are wrapped by the cameraHandler module ("grabber"), which allows any combination of cameras to be connected to the system. The BumbleBee2 transmits images in Format7 format and the stereo image pair is Bayer-tiled; therefore each stereo image pair has to be de-interlaced and transformed to a usable format (e.g. IPL image) before they are used by the image processing modules.

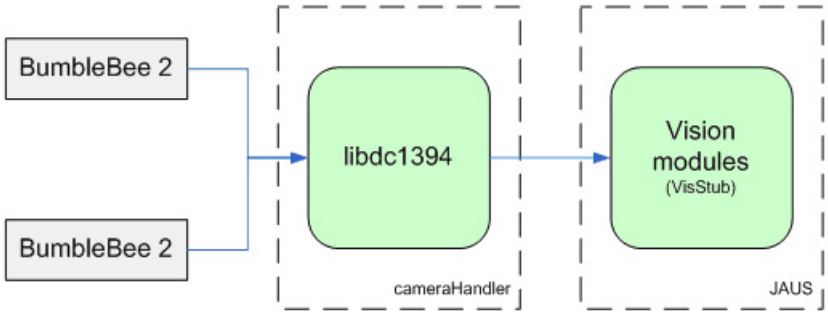


Figure 3.3: Camera software interface.

3.2 Overview of vision system

We propose a vision-based road extraction system, which uses a binocular color camera on-board and has the capability to work on urban and rural roads under dynamic lighting conditions. The road can be extracted with a wide range of road color and lighting conditions. Shadows on the road are dealt with in a manner such that it would not give the false perception of a dead-end road.

The structure of our visual system is shown in Figure 3.4. The input device is a binocular Bumblebee2 camera. It is mounted on the vehicle, pointed forward and tilted down so that it can capture images in the 5 to 50-meter range in front of the vehicle. Road extraction is accomplished by stereo processing, color training, followed by color segmentation on a pair of stereo images. In our implementation, the right image is the base image where the stereo classification and color segmentation are applied as the reference coordinates in Bumblebee2 are associated with the right image.

First, the Bumblebee captures a pair of stereo images of the road and passes them to the stereo processing module. After stereo classification, the images can be classified into ground and non-ground patches. For color sample collection, we define a trapezoidal learning region in front of the vehicle, approximately in the range from 3 to 8 meters ahead of the vehicle. In this training region, we extract the sample pixels for constructing Gaussian models after verifying those samples are from neither obstacles nor green vegetation.

Next, from the sample pixels, we construct the road color model, in a new color space. Our color model is a Gaussian mixture in an illumination-invariant color space with a variable number of Gaussians. The number of new Gaussians changes with different road conditions. In the third step, the new model is integrated into the previously constructed color model following an update rule. In the fourth step, the rest of the image is classified in the new color space to find the road surface using the updated road color model. Finally, post-processing steps follow to enhance the classified results.

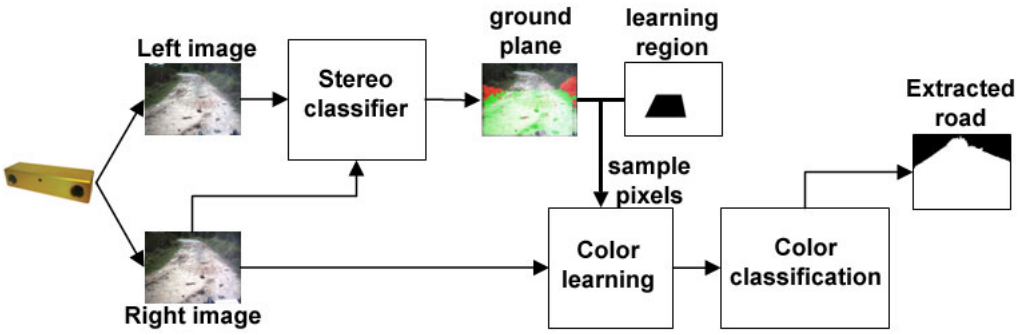


Figure 3.4: System overview.

3.3 System output specifications

The classified images can be projected into a top-view grid-map or used directly to steer the vehicle, depending on purposes or navigation algorithm. In our project, the extracted road is to be projected to a map of 225 × 75 grids (Figure 3.5), corresponding to an area of 45 m × 15 m. The road map is extended from 5 meters to 50 meters away from the vehicle. Similarly, the short-range obstacle detection result is also to be projected to an obstacle map of 30 × 40 grids, extending from 4 meters to 10 meters away from the vehicle. Figure 3.6 shows the sensing coverage of the two modules.

Figure 3.5: Process flow of long-range road extraction module.

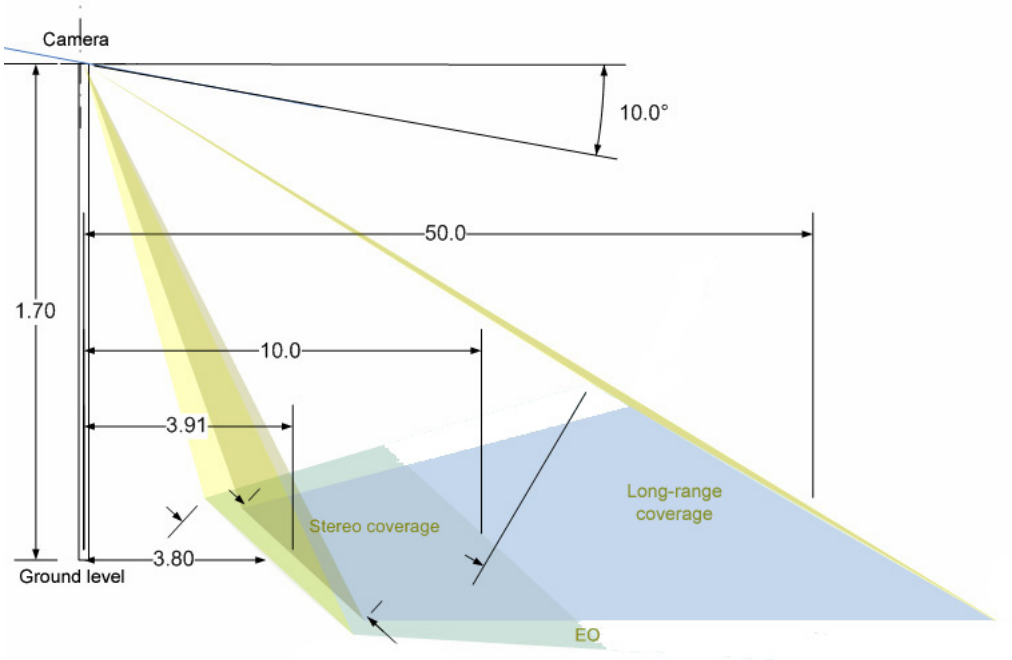


Figure 3.6: Coverage of short-range stereo and long-range road extraction.

In the projection from the classified image to the road map, a planar to planar transformation matrix is calculated. This method is known as 2D

homography [21]. It involves finding a matrix, **H** that transforms points from the image (x, y) to its corresponding 2D point on the road map (which has no height information), as shown in Figure 3.7. Though four points are needed for calculating approximately this transformation matrix, more points will lead to a more accurate transformation matrix. The selected points should cover a large area across the image because only pixels within the boundary of the selected points are transformed accurately.

The methodology of obtaining the matrix **H** requires normalization of the coordinates, then calculating **H** by singular value decomposition (SVD). Figure **??** shows the reference base image taken from the right lens of the BumbleBee2 camera. The physical ground truth is taken by directly measuring the distance with reference from the right lens of the BumbleBee2 camera. Due to resource and space constraints, only the extreme left positions can be measured for the 40m and 50m mark. From this, the homography transformation matrix can be estimated.

In this thesis, we will only discuss the first step in Figure 3.5, which is to extract the road region from the original image. Section 6.1 shows a sequence of road images with top-view road map outputs using the method presented above.

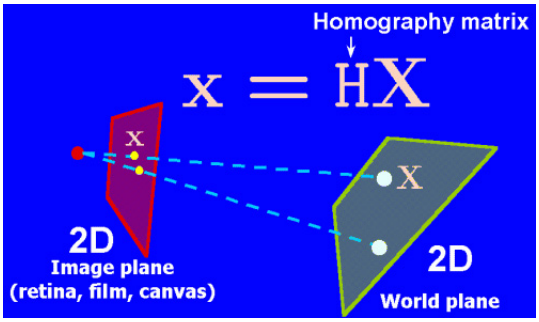


Figure 3.7: Projection from image to road map, using homography transform.

Chapter 4

Short-range Obstacle Detection

4.1 Overview

In this step, a pair of stereo images are fed to the stereo processing module for ground plane detection. This is the only step that involves the left image of the stereo pair. After the stereo disparity image is obtained, only the base image, i.e., the right image, is used in the following steps. The ground plane up to ten meters in front of the vehicle is detected. In previous techniques such as [7], the ground plane is detected by laser sensors and projected onto the color image to find the ground pixels in the image. Such approaches would involve another sensor module with sensor devices and processing software. In addition, they also need a coordinate transformation step between the sensors, which requires precise relative pose information. In this system, the final stereo classification is performed on the same base image that color segmentation is later carried out. Thus, neither relative pose information nor coordinate transformation is required.

4.2 Stereo algorithm

4.2.1 Generating cloud points

We start by computing a disparity image at 160×120 resolution with surface validation. The corresponding matching and disparity computation is performed

using the Triclops stereo library provided with Bumblebee2 camera [30]. Surface validation is enabled to improve the overall disparity output. It is a method to validate regions of a disparity map to ensure that they belong to a likely physical surface in the image. In this method, the disparity image is segmented into connected regions, and any region with an area less than a threshold is removed. The different processing stages provided by the Triclops SDK are summarized below:

1. Low-pass filtering to prepare the image for rectification. This smooths the images so that the rectification step can generate an output image with fewer aliasing effects. The low-pass filter is a 5×5 Gaussian filter.

2. Rectification of both left and right images from the same camera. This is the process of correcting for lens distortion in the input images. It also facilitates the subsequent corresponding matching process, as the images will be rectified in such a way that the rows of the left image are aligned with those of the right image. Therefore, the corresponding search is performed along the same row, effectively reducing the 2D search into a 1D search.

3. The correspondence between the stereo image pair is established by the sum of absolute intensity differences (SAD) of all the pixels within the window search space of a pair of points between the left and right images. SAD search attempts to compute the optimal disparity by minimizing the cost function

$$\min_{d=d_{min}}^{d_{max}} \left(\sum_{i=-\frac{m}{2}}^{i=\frac{m}{2}} \sum_{j=-\frac{m}{2}}^{j=\frac{m}{2}} |I_{right}[x+i][y+j] - I_{left}[x+i+d][y+j]| \right) \quad (4.1)$$

   where d_{min} and d_{max} are the minimum and maximum search disparities, and m represents the size of correlation window.

4. Surface validation attempts to find a connected region within the disparity map generated. A range of disparity values is set so that only connected pixels that lie within this range are retained because they are likely to be from the same physical object.

5. An edge map is obtained for the edge validation step, which allows correspondence between the stereo image pair to be better established.

After obtaining the correct disparity, we can perform 3D reconstruction. For each pixel at (x, y) in the disparity image with disparity d, we can compute 3D coordinates with respect to camera-centered coordinates (X_c, Y_c, Z_c) and vehicle-centered coordinates (X_w, Y_w, Z_w) as follows:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \frac{b}{d} \times \begin{bmatrix} x \\ y \\ f \end{bmatrix}, \tag{4.2}$$

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \mathbf{H} \times \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{bmatrix} \times \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}, \tag{4.3}$$

where b is the stereo baseline, f is the focal length of a camera. Additionally, $\mathbf{H}$ is a 4×4 transformation matrix, consisting of a 3×3 rotation matrix $\mathbf{R}$ and a 3×1 translation matrix $\mathbf{T}$, representing the relative camera pose in the world coordinate system. The $\mathbf{R}$ and $\mathbf{T}$ matrices are retrieved and computed through a calibration procedure [3].

After all correspondence matches in the stereo pair are 3D-reconstructed, we effectively have a 3D point cloud that represents object points in the scene. Some of those points belong to the ground while others belong to obstacles.

4.2.2 Determining ground plane

Camera pose relative to the ground is unstable during vehicle motion because of vehicle vibration and possibly slant ground. Therefore, the recovered height Z_w of an object point is not reliable for determining whether a pixel (x, y) in an image is a ground point.

A robust technique called RANSAC [19] is proposed to estimate the ground plane [1]. In this approach, we assume that most of the reconstructed 3D points (more than 50%) are ground points. Given n reconstructed points,

we draw m random sub-samples of $p = 3$ different 3D points. For any $p = 3$ non-collinear 3D points, we can determine a unique plane equation $\mathbf{P_J} = \{a_J, b_J, c_J, d_J\}$ passing these three points. If all $p = 3$ are ground points, we have a ground plane. However, as mentioned above, the problem is that we do not know whether a point is ground point, and even when all the 3 points are ground points, we do not know whether the resulting plane is the optimal ground plane that encloses the majority of ground points. Therefore, to evaluate a candidate ground plane equation $\mathbf{P_J}$, we count the number of points within the error boundaries of the plane $\mathbf{P_J}$. Intuitively, the best ground plane must have the largest number of points within its error boundaries, since the majority are ground points. Additionally, the greater the number of non-ground points (outliers), the less likely that all $p = 3$ points are *good* ground points, and, therefore, the greater the number of m random trials that should be taken.

Given our camera field of view and our specified stereo ten-meter range, the RANSAC assumption that most of the recovered 3D points are ground points is valid. In [1], the number of required trials m is dependent on the maximum fraction of non-ground points (outliers) allowed. However, for outdoor road scenes, we observed that the actual fraction of outliers is usually much lower than the maximum allowed fraction. Thus, to improve performance, we maintain a variable m number of required trials to be updated depending on the current data and a fixed m_{max} to represent the worst case. The fitting process stops when the number of trials reaches any of the two. In this way, we can have faster performance on average while still having robust and in-time performance in the worst case. The ground fitting algorithm is shown in Algorithm 1.

On our vehicle, for safe navigation, we can assume that obstacle points are at most 30% of 3D points and objects with height greater than 10cm are considered obstacles. Therefore, the height tolerance is $h_T = 0.1$ and m_{max} is computed by:

$$m_{max} = \frac{\log(0.01)}{\log(1 - (1 - 0.3)^3)} \simeq 11 \tag{4.7}$$

Ground planes that are too slanted will be rejected and the system will issue a warning message to the vehicle controller. Points within a distance of h_T from the plane are classified as ground points. In our project, the vehicle is travelling at

Algorithm 1 Ground Plane Fitting Algorithm

Require: n 3D points, maximum number of trials m_{max}, height tolerance h_T

1: Initialize counter of trials $count = 0$, best score points $s_{best} = 0$ and required number of trials $m = 1$.
2: **repeat**
3: Select three random 3D points (X_{w1}, Y_{w1}, Z_{w1}), (X_{w2}, Y_{w2}, Z_{w2}), (X_{w3}, Y_{w3}, Z_{w3}) from n points. Verify they are not collinear.
4: Construct a plane hypothesis J with normalized plane parameters $\mathbf{P_J}$=$\{a_J, b_J, c_J, d_J\}$ from three points.
5: Determine the score point s_J of the hypothesis plane J by counting the number of 3D points that are within h_T from the plane.

$$s_J = \sum_{i=1}^{n} U(h_T - h(\mathbf{p_i}, \mathbf{P_j})) \tag{4.4}$$

$$h(\mathbf{p_i}, \mathbf{P_j}) = |a_J x_i + b_J y_i + c_J z_i + d_J|, \tag{4.5}$$

where $U()$ is the unit step function.
6: **if** $s_J > s_{best}$ **then**
7: Update the best hypothesis plane and s_{best}
8: Update the required attempts m.

$$m = \frac{\log(0.01)}{\log(1 - \frac{s_{best}}{n}^3)} \tag{4.6}$$

9: **end if**
10: Increment counter of trials: $count$.
11: **until** $count > m$ OR $count > m_{max}$.
12: Analyze the best plane to return status of the ground plane fitting process: good, slant ground, no ground, etc.

a relative high speed of more than 20 m/s. Therefore, the ground plane changes every frame, and it is necessary to re-estimate the ground plane at every cycle.

After the ground plane is determined, we define a trapezoidal learning region in the image, approximately a small area in front of the vehicle. Only color pixels in the learning region that are validated as ground by the stereo vision are extracted for Gaussian model construction. Fig. 4.1 shows the stereo-classified result and learning region position in the image.

Figure 4.1: Learning region (black trapezoid) and detected ground plane (tinted green) from a pair of stereo images (top images).

4.3 Color sample collection

In our multi-range architecture, the short-range module is used to provide learning samples for the long-range road extraction module. Such an arrangement would make the system adaptive to different driving environments as the vehicle moves. In this section, the sample collection method will be presented. Essentially, the road color samples will be collected from a small training area right in front of the vehicle. To ensure that those color samples are not wrongly collected from obstacles or green vegetation accidentally inside the training area and lead to an incorrect road color model, obstacle and green vegetation in the training

area will be removed. Stereo results in the previous section will be utilized to remove the obstacle, while some fast classification methods are used to remove vegetation in the training area.

4.3.1 Training area

The training area is a small area defined in the image. Since road color samples will be collected from this area, the area must not be too small or too large. Too small a training area would cause fewer color samples to be collected and the subsequent color model would not be sufficiently representative, while too large a training area would lead to a higher chance of outliers wrongly collected as samples and possibly distort the subsequent color model. Although we have different mechanisms to remove outliers such as obstacle and vegetation from the training area, such mechanisms are more reliable if the training area is small and close to the vehicle. In our implementation, the training region is fixed in the image such that it approximately corresponds to a 4m $\times$ 4m area which is about 3-7 meters away from the vehicle, as shown in Figure 4.1.

4.3.2 Obstacle removal

The road color models are constructed based on color samples from a training region. For the road color models to be valid for correct road classification, the color samples must not be from an obstacle that is possibly inside the training region. Previous methods either assume that the training area is free of obstacle [37], or use another sensor system that very much increases system complexity [35]. In our current approach, the already obtained stereo classification output is utilized to verify areas in training region that can be used for color sample collection. This can be done by simply finding the intersection of the stereo-classified image and the training region mask image (binary AND operation).

4.3.3 Green vegetation removal

Stereo classification output has been used to remove samples from obstacles in training region. However, stereo, as a range-based method, cannot differentiate

efficiently between a dirt road and adjacent flat grassy areas. As the colors of drivable road areas and non-drivable vegetation and grassy areas are very different, color-based long range can differentiate vegetation areas efficiently provided that the road color models are correctly constructed. Road colors are learned from color samples collected from a fixed training region in front of the vehicle. However, if green grass samples in the training region are wrongly collected and assumed as a road color, it is possible that roadside vegetation would be misclassified as drivable by color-based long range module. Thus, it is crucial that all green grassy areas within training region must be removed, as they might interfere with proper color model construction. Therefore, several methods are proposed for fast and fairly reliable detection and removal of green vegetation pixels. For fast processing, vegetation removal will be performed on the intersection mask obtained in the previous step (see Section 4.3.2).

4.3.3.1 Look-up table

During early developments (see Section 5.1), it is observed when plotting a large number of color samples collected from vegetation images on Hue-Saturation histogram, they form a neat cluster in the middle of the 2D histogram, corresponding to vegetation colors (Figure 4.2). Therefore, a simple but effective look-up table method is proposed to remove grass-green color samples from color samples that are collected for road color model construction.

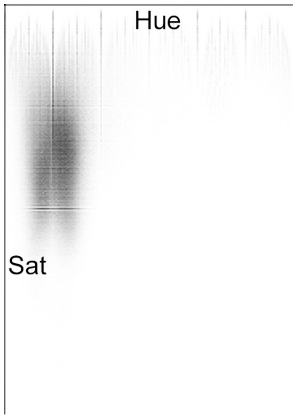


Figure 4.2: Hue-Sat 2D histograms used as look-up tables for green vegetation area. Darker point at coordinates (H,S) means higher population with value (H,S).

First we collect a number of green vegetation images. From the training images, we plot the cumulative 2D Hue-Saturation histogram and store it in a table. During color sample collection, the previously constructed histogram would be loaded as a look-up table. For each collected RGB color sample, we compute the Hue and Saturation value for that sample in HLS color space, and look for the population value at the corresponding cell in the table. Only those samples with population value below some threshold value are passed into the training phase. As this look-up table method is sensitive to noise, the training area is blurred before processing.

Experiments show that this method can remove a large number of grass-green color samples (Figure 4.3). As in the current color road classification algorithms, Gaussians with too few supporting samples would be discarded. This verification mechanism actually helps to remove green grass areas from the training region.

(a) Training area (b) Original image (c) Classified image

Figure 4.3: Green vegetation removal using look-up table. Grass area does not get trained and model remains valid.

4.3.3.2 Pre-trained Gaussian mixture model of vegetation

Another alternative method for green vegetation removal is proposed, similar to classification step discussed in Section 5.3.3. Essentially, this vegetation removal method has exactly the same idea as the road extraction method (see Chapter 5), using Gaussian mixture as the color model for classification of color pixels.

However, unlike general road classification, the training area is close to the vehicle and camera, and does not suffer low brightness problem. Therefore, the classification step can be much simplified, for faster processing. In addition,

the vegetation color model is fixed and determined off-line through a number of sample images. Thus, there is no color model construction step as well as color update step. The post-processing step is also skipped as any outlier regions in the small training region are significant.

In brief, the green vegetation removal can be described as follows: for each pixel in the training region and not removed by stereo, we find the minimum Mahalanobis distance from it to the mean vectors of the pre-trained vegetation color models. The pixel is classified as vegetation if the distance is less than some threshold value. Otherwise, the pixel is non-vegetation and will be collected as color samples. The procedure to find the vegetation pixel may be summarized by the following equation:

$$d(\mathbf{p}, \mu_{vegetation})_{min} = ((\mathbf{p} - \mu_{vegetation})^T \Sigma^{-1}_{vegetation}(\mathbf{p} - \mu_{vegetation}))_{min} < d_{classify}.$$

where $\mu_{vegetation}$, $\Sigma_{vegetation}$ are mean vectors and covariance matrices of the pre-defined vegetation color model.

Similar to the previous vegetation removal approach, experiments also show that this method can remove a large number of grass-green color samples (Figure 4.4). It can be observed that the road classification results are almost identical in Figures 4.3 and 4.4 since the road color model is correctly constructed after most of the grass samples are removed.

(a) Training area (b) Original image (c) Classified image

Figure 4.4: Green vegetation removal using pre-trained Gaussian mixture. Grass area does not get trained and model remains valid.

In our experiments, we will adopt the latter method to utilize the modules and functions developed for road classification.

Chapter 5

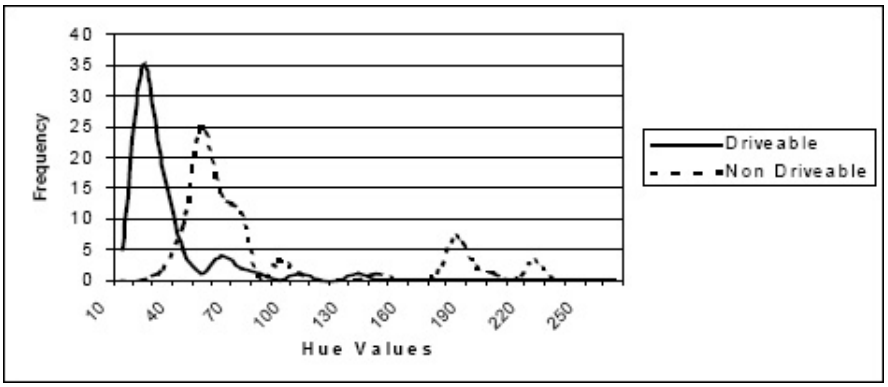
Long-range Road Extraction

5.1 Overview - Early developments and current approach

5.1.1 Linear thresholding approach

This approach is the earliest and simplest approach that has been implemented in our project. In [5], Chaturvedi et al. observed that distribution of drivable and non-drivable pixels peaks at different Hue values in their histograms, as shown in Figure 5.1(a). In addition, the overlapping area between the two histograms is small. Based on that observation, they proposed a road extraction method by thresholding at a suitable threshold Hue value. In addition, they argued that Hue as a chromatic component in HSI color space is invariant to large variations in lighting conditions throughout the day and in shaded areas.

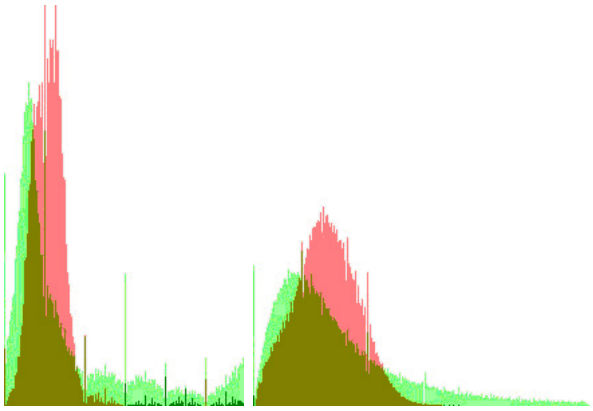
However, the above observations are only true in limited driving environments, such as mud road, shown in Figure 5.1(b). As we move to different terrains and environments, such methods will not work properly. This could be explained that in our driving environments, chromatic values (Hue and Saturation) of road and non-road areas are less distinct, as illustrated in Figure 5.2. However, we observe that in the 2D Hue-Sat histogram, the colors of road and non-road areas are still distinguishable. As shown in Figure 5.6, pixels from road and non-road sample images are concentrated in two different clusters, and they can be separated by a single straight line.



(a) Histogram of Hue values for drivable and non-drivable areas

(b) The target driving environments

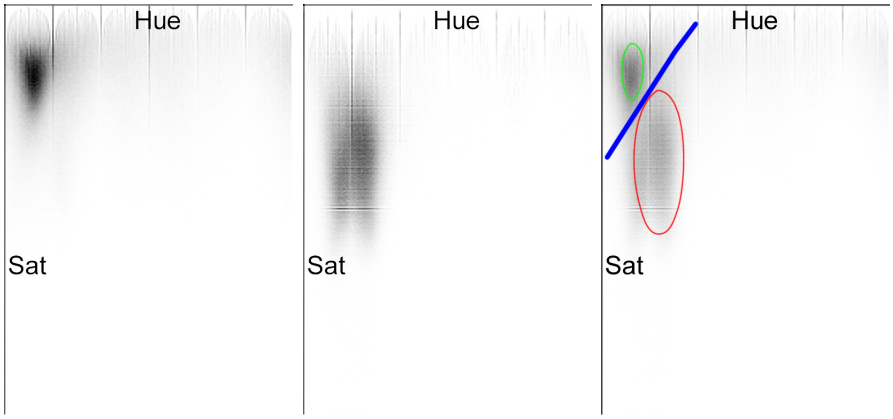
Figure 5.1: Road extraction method by linear thresholding, by Chaturvedi [5].



(a) Histogram of Hue values

(b) Histogram of Saturation values

Figure 5.2: Hue and Sat histograms for drivable (green) and non-drivable (red) areas.



(a) Hue-Sat histogram of road samples

(b) Hue-Sat histogram of non-road samples

(c) Combined histogram with estimated separation line

Figure 5.3: Hue-Sat 2D histogram for drivable and non-drivable areas. Darker point at coordinates (H,S) means higher population with value (H,S). The estimated line (blue) separates drivable (green) and non-road (red) clusters.

Therefore, we perform thresholding by using both Hue and Sat values instead of only Hue values in the original approach [5]. We assume that for incoming images, any pixel in drivable and non-drivable areas would have a tendency to be in corresponding regions in the Hue-Sat histogram. The classification of road and non-road pixels is determined by the following linear equation:

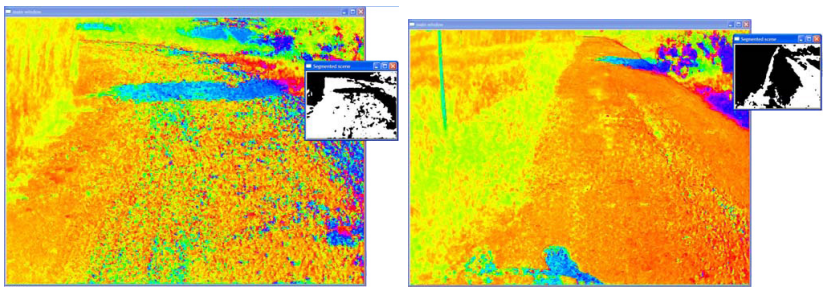
$$A * \text{Hue} + B * \text{Sat} + C \begin{cases} > 0 \Rightarrow \text{road} \\ \leq 0 \Rightarrow \text{non-road} \end{cases} \tag{5.1}$$

where A, B, C are determined off-line through histogram analysis of sample road and non-road images. Hue and Saturation in HSI color space are computed using the formulas in Table 5.1.

The method was tested in different driving environments with moderate success. Although it can generally extract road areas correctly, it will misclassify once the color is slightly different (Figure 5.4). As the clusters in the Hue-Sat histogram only represent the majority of the road and non-road color pixels, pixels with different colors may be misclassified. In addition, a single straight line as the boundary between road and non-road in the Hue-Sat histogram may not be adequate (Figure 5.5). It is quite probable that road areas have more than one major color. Thus, the boundary between road and non-road pixels should be more complex than just a single straight line. Our experiments show

Table 5.1: Conversion from RGB color space to HSI color space

Given R, G, B values scaled to (0..1) range
$V_{max} \leftarrow \max(R, G, B)$
$V_{min} \leftarrow \min(R, G, B)$
$L \leftarrow \frac{V_{max}+V_{min}}{2}$
$S \leftarrow \begin{cases} \frac{V_{max}-V_{min}}{V_{max}+V_{min}} & \text{if } L < 0.5 \\ \frac{V_{max}-V_{min}}{2-(V_{max}+V_{min})} & \text{if } L \geq 0.5 \end{cases}$
$H \leftarrow \begin{cases} \frac{(G-B)\times 60}{S} & \text{if } V_{max} = R \\ 180 + \frac{(B-R)\times 60}{S} & \text{if } V_{max} = G \\ 240 + \frac{(R-G)\times 60}{S} & \text{if } V_{max} = B \end{cases}$
if $H < 0$ then $H \leftarrow -H + 360$
For 8-bit representation: $H \leftarrow H/2$

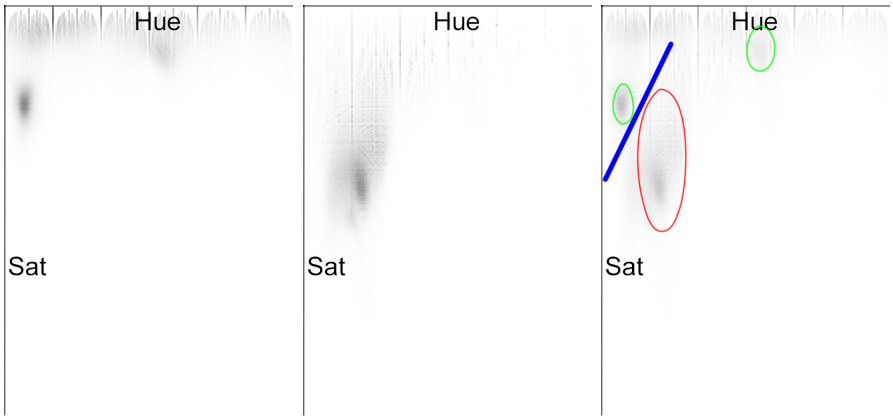


(a) Area on road with dissimilar color

(b) Road section with dissimilar color

Figure 5.4: Misclassified results by linear thresholding approach. Hue color (large window) and classified output (small window).

that this road extraction method is not a robust and complete solution for road extraction.



(a) Hue-Sat histogram of road samples

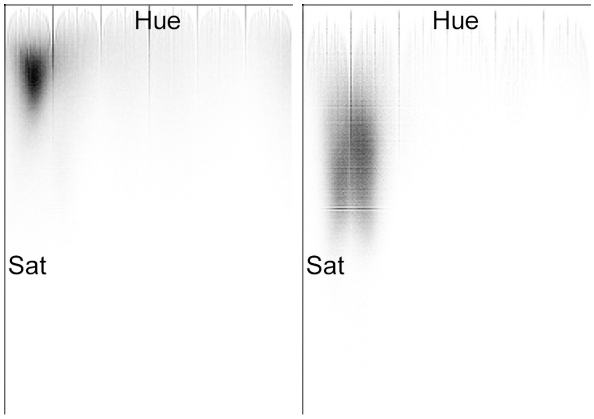
(b) Hue-Sat histogram of non-road samples

(c) Combined histogram with estimated separation line

Figure 5.5: Weakness of linear thresholding approach. A single line (blue) could not separate drivable (green) and non-road (red) clusters.

5.1.2 Look-up table approach

In the linear thresholding approach, some driving environments show that a single straight line boundary is not sufficient to effectively separate drivable and non-drivable pixels in the Hue-Sat histogram. Thus, we proposed another approach, called the look-up table (LUT) approach. In this approach, 2D Hue-Sat histograms for road and non-road sample images are stored as two 2D LUTs for later reference. As a histogram, each cell in a table contains the population value for a particular Hue and Saturation value (H,S). Therefore, cells with (H,S) values corresponding to dominant colors of road or non-road will generally have higher population values. During the classification step, for each pixel in the image, we will look up in the road and non-road tables the population values, using the Hue value and Saturation value of that particular pixel as coordinates. If the population value from the road table is larger than that of the non-road table, the pixel is classified as road; otherwise, it is non-road. In this way, LUT approach will allow finer separation between road and non-road pixels on the histogram.



(a) Look-up table for road

(b) Look-up table for non-road

Figure 5.6: Hue-Sat 2D histograms used as look-up tables for road and non-road. Darker point at coordinates (H,S) means higher population with value (H,S).

The confidence of classification is determined based on the difference in percentage values. As the LUT approach is sensitive to noise, the image is blurred before processing. In our implementation, the tables are constructed off-line. First, a number of road and non-road sample images are collected. Then, two cumulative Hue-Sat histograms for road and non-road images are constructed and stored as tables. During processing, these tables will be loaded and referred to for the population values.

In general, the LUT approach performs better than linear thresholding approach (Figure 5.7). However, there are still some limitations. First, at the near range, the rural and jungle roads are often filled with colored stones and outliers. Even though the image is blurred in pre-processing to remove those outliers, the outcome is unpredictable at the near range. Besides, the classification is correct only in moderate lighting conditions. In the more extreme lighting conditions, such as around noon time, the classification is less stable (Figure 5.8). This could be explained that Hue and Saturation values, which are supposed to be invariant to lighting, are not really invariant when the brightness is too high or too low. This fact was explained theoretically in Section 2.2.2.1. Finally, the LUT approach does not provide a general solution to road extraction problem. This classification method only works in limited environments that road and non-road samples have been collected from. When moving to another driving

environment, such as from one road section to another with a different road color, the method will fail. This is undesirable since keeping different LUTs for different road sections will increase system complexity. Also, it is not feasible to predict all possible driving terrains, collect the samples and construct the tables. Therefore, this LUT approach has been shown not to be a robust and complete solution for road extraction.

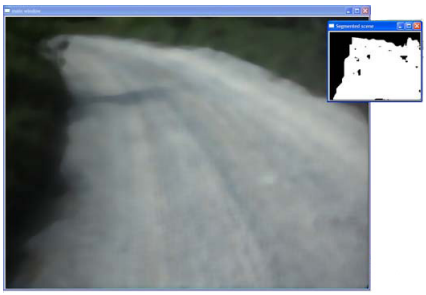
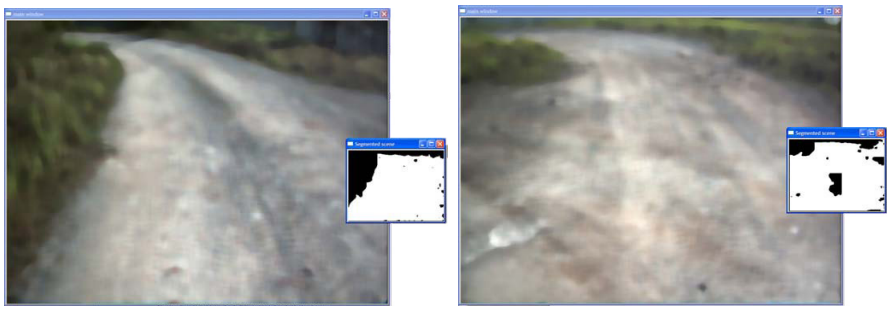


Figure 5.7: Look-up table classification result.



(a) Misclassified vegetation (b) Misclassified road

Figure 5.8: Weaknesses of LUT approach.

5.1.3 Current approach

In the current approach, we propose a novel road extraction method that overcomes the above limitations, namely limited working environment and degraded performance against lighting changes. The above off-line approaches apparently do not reflect how humans detect road and non-road areas when driving. A human has no exact memory of colors of the previously driven roads in the past for differentiating road and non-road areas. Rather, we learn the color of the road from areas in the vicinity of our current location, and proceed to find further

areas with similar color. As we move along the road, we keep updating the color of the road such that when the vehicle moves to another terrain with a different color, we quickly learn, adapt to the new road color and continue to find areas with color similar to the new color.

In our current approach, we adopt that view into color-based road extraction module. A multi-range architecture is proposed, in which road color is learned from near-range for classification at farther range. Color sample collection was presented in Chapter 4. From road color samples collected at near range, a road color model is constructed. A Gaussian mixture is used to represent the road color model. This color model will be updated as the vehicle moves, making the method adaptive to different driving environments. In addition, to cope with lighting changes and shadows, the color model construction and color classification will be performed in a new color space that is invariant to illumination, instead of RGB color space. We will first discuss the new illumination-invariant color space and its conversion from RGB in Section 5.2. We then present our color learning and updating mechanism in Section 5.3.

5.2 Color conversion

5.2.1 Derivation of conversion formula

As discussed in Section 2.2.1, a digital color image is an array of pixels with each pixel denoting the incoming light signal's intensity received at the image sensor. This light intensity is determined by two components: the first component depends on the colors and intensity of the illuminant, and the second component depends on the reflectance properties of the illuminated surface. In this section, we aim to remove the illuminant component and find a measurement such that it is representative of the reflectance component.

Color images captured from a conventional camera would have three separate red, green, and blue (RGB) channels. As shown in Subsection 2.2.1.3, the intensity of each channel is described by:

$$\Phi_k = \int E(\lambda)S(\lambda)Q_k(\lambda)d\lambda,\ k = R, G, B. \tag{5.2}$$

where $E(\lambda)$ is the illumination spectral power distribution, $S(\lambda)$ is the surface's spectral reflectance distribution function, and $Q_k(\lambda)$ is the spectral sensitivity function of the sensor for each channel.

Suppose that the image sensors' spectral sensitivity functions are narrow-band such that they can be approximated by a Dirac delta function $Q_k(\lambda) = q_k\delta(\lambda - \lambda_k)$, where q_k represents the sensor strength. Using this approximation, Equation (5.2) will be simplified to:

$$\Phi_k = q_k E(\lambda_k)S(\lambda_k),\ k = R, G, B, \tag{5.3}$$

In addition, it is shown that natural daylight has the color temperature of approximately 6500 K, and, therefore, its illumination distribution function $E(\lambda)$ can be approximated by Planck's law:

$$E(\lambda, T) \propto \frac{2hc^2}{\lambda^5}\frac{1}{\exp\frac{hc}{\lambda kT} - 1}, \tag{5.4}$$

where T is the temperature of the black body in Kelvin degrees, λ is the wavelength, and h, k, c are Planck's constant, Boltzmann's constant, and light speed constant, respectively.

Given the temperature range 3000-7000 K of conventional outdoor illuminants, and the wavelength range 400-750 nm of the visible spectrum, we have:

$$\exp\frac{hc}{\lambda kT} \simeq \exp\frac{6.63 \times 10^{-34} \times 3.00 \times 10^8}{1.38 \times 10^{-23} \times 6.5 \times 10^3 \times 500 \times 10^{-9}} = \exp 4.43 = 84.33 \gg 1.$$

Therefore, we can further approximate the illumination distribution function by:

$$E(\lambda, T) = Ic_1\lambda^{-5}(\exp\frac{c_2}{T\lambda} - 1)^{-1} \approx Ic_1\lambda^{-5}\exp\frac{-c_2}{T\lambda}, \tag{5.5}$$

where c_1, c_2 are constants and I represents the intensity of the incident light. Substituting into Equation (5.3), we have the approximate sensor response function:

$$\Phi_k = Ic_1\lambda_k^{-5}\exp\frac{-c_2}{T\lambda_k}S(\lambda_k)q_k = Is_k\exp\frac{i_k}{T}, \tag{5.6}$$

with $s_k = c_1\lambda_k^{-5}S(\lambda_k)q_k$, $i_k = -c_2/\lambda_k$. The logarithm of the sensor responses for the three channels can be represented by:

$$\log\Phi_k = \log I + \log s_k + \frac{i_k}{T}$$

$$\Rightarrow \begin{bmatrix} \log \Phi_R \\ \log \Phi_G \\ \log \Phi_B \end{bmatrix} = \begin{bmatrix} \log I & \log s_r & \frac{i_r}{T} \\ \log I & \log s_g & \frac{i_g}{T} \\ \log I & \log s_b & \frac{i_b}{T} \end{bmatrix} = \begin{bmatrix} 1 & i_r & \log s_r \\ 1 & i_g & \log s_g \\ 1 & i_b & \log s_b \end{bmatrix} \begin{bmatrix} \log I \\ \frac{1}{T} \\ 1 \end{bmatrix} \tag{5.7}$$

$$\Rightarrow \begin{bmatrix} \log R \\ \log G \\ \log B \end{bmatrix} = \begin{bmatrix} 1 & i_r & \log s_r \\ 1 & i_g & \log s_g \\ 1 & i_b & \log s_b \end{bmatrix} \begin{bmatrix} \log I \\ \frac{1}{T} \\ 1 \end{bmatrix} \tag{5.8}$$

where the sensor responses (Φ_R, Φ_G, Φ_B) are corresponding to the RGB values in the color image. To retrieve m illumination-invariant measurements ζ_i from (R, G, B) colors, we have an $m \times 3$ conversion matrix $\mathbf{Z}$ such that:

$$\begin{bmatrix} \zeta_1 \\ ... \\ \zeta_m \end{bmatrix} = \mathbf{Z} \begin{bmatrix} \log R \\ \log G \\ \log B \end{bmatrix} = \mathbf{Z} \begin{bmatrix} 1 & i_r & \log s_r \\ 1 & i_g & \log s_g \\ 1 & i_b & \log s_b \end{bmatrix} \begin{bmatrix} \log I \\ \frac{1}{T} \\ 1 \end{bmatrix}$$

In Equation (5.8), I and T are changing and dependent on illuminant. Therefore, to remove illumination dependence, we must have

$$\mathbf{Z} \times \begin{bmatrix} 1 & i_r & \log s_r \\ 1 & i_g & \log s_g \\ 1 & i_b & \log s_b \end{bmatrix} = \begin{bmatrix} 0 & 0 & \zeta_1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & \zeta_m \end{bmatrix} \tag{5.9}$$

Therefore, each row vector $\mathbf{r} = (r_1, r_2, r_3)$ of $\mathbf{Z}$ would be a non-trivial solution of the following homogeneous equations:

$$\mathbf{r} \times \begin{bmatrix} 1 & i_r & \log s_r \\ 1 & i_g & \log s_g \\ 1 & i_b & \log s_b \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} \begin{bmatrix} 1 & i_r & \log s_r \\ 1 & i_g & \log s_g \\ 1 & i_b & \log s_b \end{bmatrix} = \begin{bmatrix} 0 & 0 & \zeta_i \end{bmatrix}$$

$$\Rightarrow \begin{cases} r_1 + r_2 + r_3 & = 0 \\ i_r r_1 + i_g r_2 + i_b r_3 & = 0 \end{cases}$$

$$\Rightarrow \begin{bmatrix} 1 & 1 & 1 \\ i_r & i_g & i_b \end{bmatrix} \mathbf{r}^T = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5.10}$$

Theoretically, Equation (5.10) has only one linearly independent general solution $\mathbf{r_0}$. However, experiments show that classification in such 1D intrinsic image

would lead to erroneous results. It comes from the fact that the sensor response Φ_k must be in some value range for the approximations in Equations (5.2)-(5.8) to be valid. Especially, the assumptions of intrinsic color space are certainly invalid for the dark areas in the image, as discussed later in Section 5.3.3.

Meanwhile, experiments also show that by using an additional solution for Equation (5.10) and calibrating independently, we will have a more stable 2D representation. Observations on many experiments indicate that 2D classification is more resistant to errors than 1D classification, especially at dark areas in the image, as shown in Fig. 5.9. Therefore, we utilize $m = 2$ solutions for more stable classification:

$$\mathbf{Z} = \begin{bmatrix} \cos\alpha & \sin\alpha & -\cos\alpha - \sin\alpha \\ \cos\beta & -\cos\beta - \sin\beta & \sin\beta \end{bmatrix}, \tag{5.11}$$

$$\begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix} = \begin{bmatrix} \log\frac{R}{B}\cos\alpha + \log\frac{G}{B}\sin\alpha \\ \log\frac{R}{G}\cos\beta + \log\frac{B}{G}\sin\beta \end{bmatrix}, \tag{5.12}$$

where $\alpha = \arctan -\frac{i_r - i_b}{i_g - i_b}$ and $\beta = \arctan -\frac{i_r - i_g}{i_b - i_g}$.

As discussed in the next section, the angles α and β cannot be theoretically derived and has to be calibrated manually, with limited resolution (0.1° resolution) and limited accuracy. Because α and β are calibrated independently, the two measurements ζ_1 and ζ_2 are not completely dependent, although highly correlated. Complete dependency only exists when the angles α and β are retrieved with high accuracy (near theoretical values). In addition, given the low resolution of the calibration process, having two invariant measurements will limit the calibration error and improve the confidence of illumination variance.

Figure 5.9: Results from road classification (tinted red) in 2D intrinsic colors (ζ_1, ζ_2) (left) and 1D intrinsic color (ζ_1) (right).

5.2.2 Camera calibration

In pratice, the above α and β formulas are not helpful in determining the two angles for cameras since the wavelengths λ_k are unknown and different on different cameras. Therefore, we need to perform off-line camera calibration to find the values of the angles. The two angles are determined separately and independently to avoid error accumulation.

In our implementation, we calibrate a camera by using road images that were captured by it, and which had balanced shaded and non-shaded areas (Figure 5.10). We experimented with different angles and searched for the best angle which gave consistent values for all the road regions. The consistency can be analyzed visually and roughly measured numerically by their entropy values. The process of finding the best angles numerically is summarized in Algorithm 2. The process is repeated on several images to find the best angles. Experiments show that the best angles are usually close to the minimum-entropy angles found from Algorithm 2, but seldom exactly at those angles. Therefore, for fine search of the best angles, the resultant intrinsic images should be visually inspected. For our Bumblebee2 camera, the angles are found as $\alpha = 119°$ and $\beta = 41.1°$. Figure 5.10 shows shady road scenes captured by the Bumblebee2 camera and their corresponding intrinsic color values ζ_1, ζ_2.

5.3 Color classification

The outline of the color-based road extracting process is illustrated by the flowchart in Fig. 5.11.

5.3.1 Gaussian color model construction

After pixel sampling, we construct the road color models which are Gaussian mixture models. In contrast to previous techniques such as [6] [7] [34] with a fixed number of learned models, a flexible number of models are learned from the training samples. The optimal number of models to represent road colors depends on road conditions. Generally, badly-maintained rural roads require a higher number of models while tarmac roads require fewer models. By training

Algorithm 2 Camera calibration procedure

Require: Road images in RGB colors with balanced shaded and non-shaded regions.

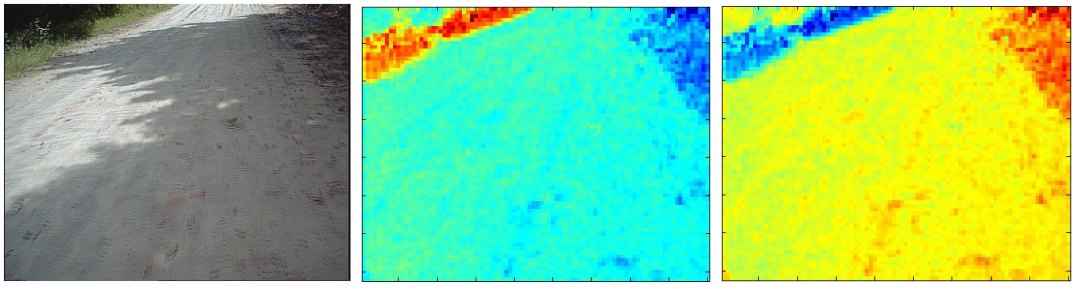
1: **for** each road image **do**
2: **for** θ_1, $\theta_2 = 0°$ to $180°$ **do**
3: Compute two independent gray-scale images from original RGB image

$$\begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix} = \begin{bmatrix} \log\frac{R}{B}\cos\theta_1 + \log\frac{G}{B}\sin\theta_1 \\ \log\frac{R}{G}\cos\theta_2 + \log\frac{B}{G}\sin\theta_2 \end{bmatrix}. \tag{5.13}$$

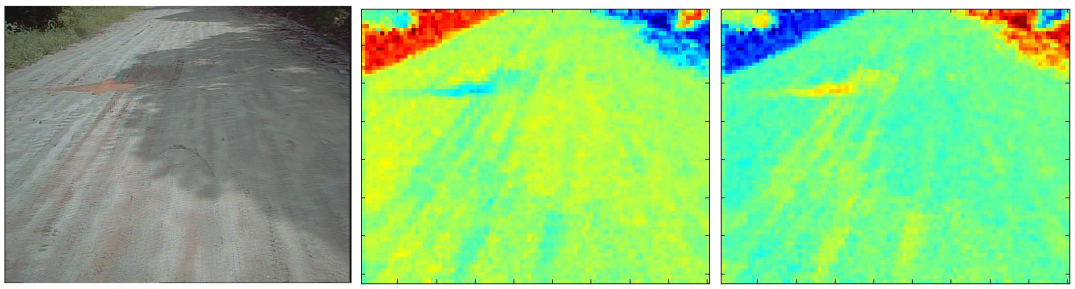
4: Find pixels in top and bottom 5% of value ranges and remove (to reduce noise).
5: Calculate bin width using Scott's Rule [31]

$$h = 3.49\text{std}(\zeta_k)N^{-1/3}, k = 1, 2. \tag{5.14}$$

6: Construct histograms for gray-scale images.
7: Compute entropy values from the histograms.
8: Keep track of minimum entropy values and corresponding angles.
9: **end for**
10: **end for**
11: Return minimum-entropy angles: $\alpha = \theta_1^{min}$, $\beta = \theta_2^{min}$.



(a) Original RGB image (b) ζ_1 with $\alpha = 119°$ (c) ζ_2 with $\beta = 41.1°$



(d) Original RGB image (e) ζ_1 with $\alpha = 119°$ (f) ζ_2 with $\beta = 41.1°$

Figure 5.10: Road scenes with shadows and corresponding intrinsic images.

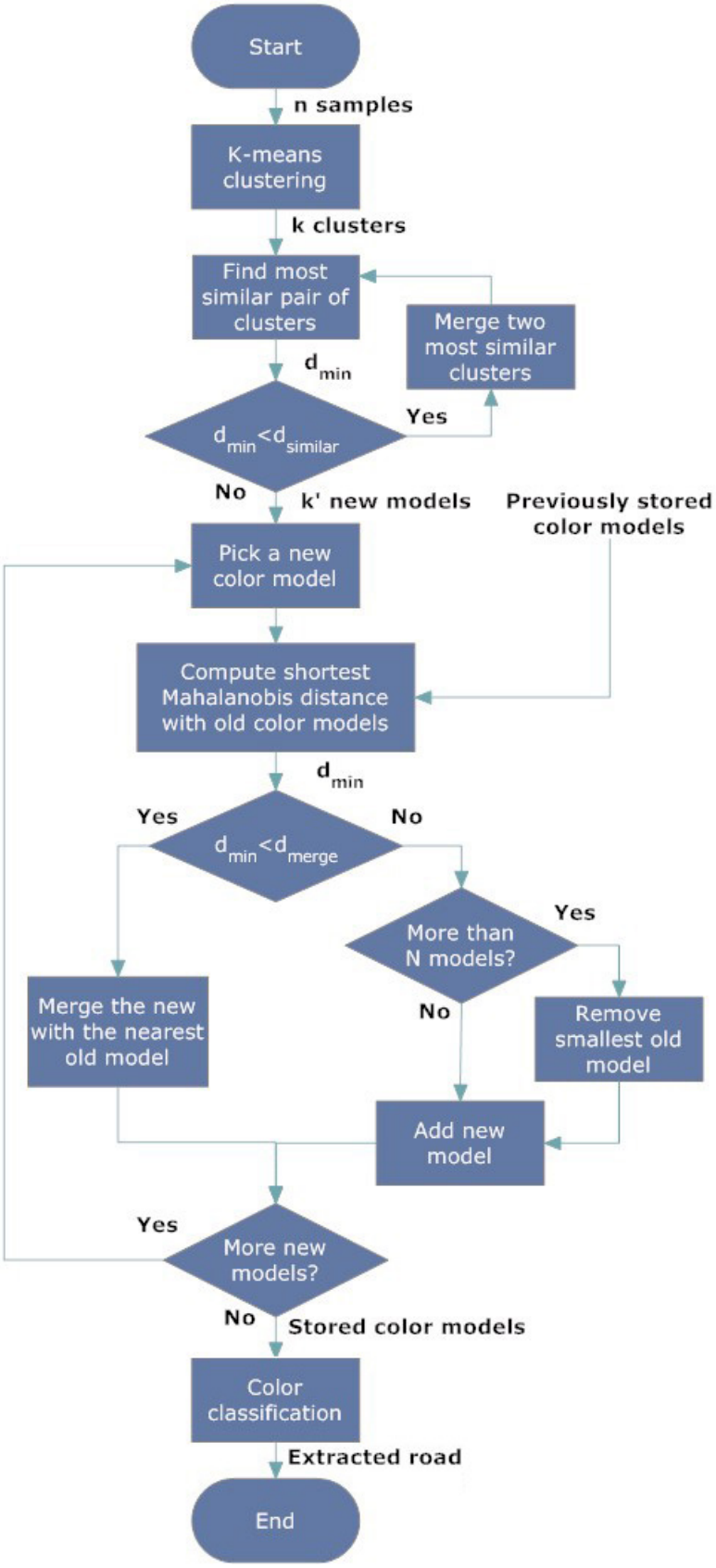


Figure 5.11: The workflow diagram of the color-based road extraction algorithm.

a variable number of models, we can avoid the over-fitting problem when the number is too high or erroneous segmentation when the number is too low.

From n collected sample, we fit them into k clusters using K-means clustering where k is sufficiently large. Then, each cluster is characterized by a mean vector μ_c, a covariance matrix Σ_c, and a mass number m_c. The mass number is the number of pixels in each cluster and the mean vector is the average of the cluster's samples. The covariance matrix is the same for all clusters and equals to Σ_0, as computed from all training samples:

$$\mu_{\mathbf{c}} = \frac{\sum_{i=1}^{m_c} \mathbf{p}_i}{m_c}, \tag{5.15}$$

$$\Sigma_c = \Sigma_0 = \frac{\sum_{i=1}^{n} (\mathbf{p}_i - \mu_{\mathbf{c}})(\mathbf{p}_i - \mu_{\mathbf{c}})^T}{n}, \tag{5.16}$$

where $c = (1 \ldots k)$. The clusters are merged by agglomerative hierarchical clustering (AHC) with the similarity measure between two clusters given by:

$$d(C_i, C_j) = (\mu_i - \mu_j)^T \Sigma_0^{-1} (\mu_i - \mu_j). \tag{5.17}$$

A new model is created in place of two original models and would have the following attributes:

$$m_{merged} = m_i + m_j \tag{5.18}$$

$$\mu_{merged} = \frac{m_i \mu_i + m_j \mu_j}{m_i + m_j} \tag{5.19}$$

$$\Sigma_{merged} = \Sigma_0. \tag{5.20}$$

The merging process stops when the closest two clusters have the distance exceeding $d_{similar}$. Among the models left after merging, those models with mass number m_c less than 5% of the sample number are regarded as outliers and discarded. Finally, we have k' training models. Apparently, the initial k should be not too small to affect the converged k' or too large to affect the performance. In our implementation, we set k is the final k' in the previous frame added by a small constant c_k, on the assumption that the road colors are similar in the two frames.

5.3.2 Color model updating

After the color models are constructed from the sample pixels, they are integrated with the previously constructed color models. We keep a fixed number N of color models in the memory from previously processed frames.

If there exists one old color model i and one new model that satisfy the condition

$$d(C_i, C_j) = (\mu_i - \mu_j)^T (\Sigma_i + \Sigma_j)^{-1} (\mu_i - \mu_j) \leq d_{merge}, \tag{5.21}$$

the models are regarded as similar and merged into a new color model and its attributes are computed as follows:

$$m_{merged} = m_i + m_j \tag{5.22}$$

$$\mu_{merged} = \frac{m_i \mu_i + m_j \mu_j}{m_i + m_j} \tag{5.23}$$

$$\Sigma_{merged} = \frac{m_i \Sigma_i + m_j \Sigma_j}{m_i + m_j}. \tag{5.24}$$

There are other ways to compute the covariance matrix of the merged model. However, the above formula is simple and generally acceptable as the covariance matrices are usually similar.

If there is no such correspondence between the new and old models, the following rule applies; if the number of old models is less than N, we append the new models into empty spaces; if the number of old models is already at maximum N, we replace the old models with the smallest mass numbers with the new models. After model updating, we decrease the mass number m of each model by a decay factor. This is to insure that the old and irrelevant models after some time would become insignificant and discarded.

In some implementations such as [6] [7] [14], the color models are trained and updated every frame. However, such model update rate is computationally expensive and unnecessary. It is perceived that, given our camera's capture rate (6 Hz) and vehicle speed, the colors of the road are similar across several frames. Thus, the color models are valid for a number of frames and need not be updated in every frame. The optimal training frequency has to be empirically determined.

Experiments show that training at higher frequencies (updating after fewer frames) would generally provide less noisy outputs. Furthermore, whenever the

color models become invalid, training at lower frequencies would also lead to a delay in correcting the color models. Color models can become invalid whenever the vehicle moves from one terrain to another or just simply turn away from the sun, leading to overall camera exposure change. Based on experiments, it is therefore recommended that the optimal training frequency is to update color models once after 3-6 frames.

5.3.3 Road classification

After the road color models are constructed, we can classify the rest of the image to find the road pixels. Only models with mass number above some fraction $f_{classify}$ of the largest mass number are considered. For each pixel, we find the minimum Mahalanobis distance from it to the mean vectors of the color models. The pixel is classified as road if the distance is less than some threshold value and non-road otherwise:

$$d(\mathbf{p}, \mu_i)_{min} = ((\mathbf{p} - \mu_i)^T \Sigma_i^{-1} (\mathbf{p} - \mu_i))_{min} < d_{classify} \tag{5.25}$$

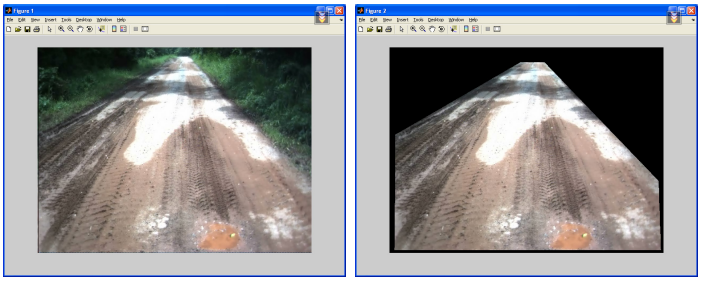
By classifying in the intrinsic color space, shadows can be classified as drivable. However, in this color space, very dark areas in vegetation are often misclassified as non-drivable (Figure 5.12(b)).

This can be explained by noting that dark colored regions may be ambiguously determined to be road or non road, especially in intrinsic color. Figure 5.14 shows a plot of distribution of the color pixels in RGB color space. A typical road image is manually segmented into road, vegetation and dark regions (Figure 5.13). Color pixels from these regions are plotted into RGB color space, with their corresponding colors which are blue, cyan & red, and pink, respectively. The plot shows that in the dark region around (0,0,0), the pixel clusters of road, vegetation and dark areas overlap significantly. Since intrinsic value is computed from RGB values, ambiguous RGB values would lead to even more ambiguous intrinsic values. This explains why a dark pixel close to (0,0,0) is so ambiguous, and can be classified as either road or non-road in intrinsic color. Experiments with other road scenes also give similar observations.

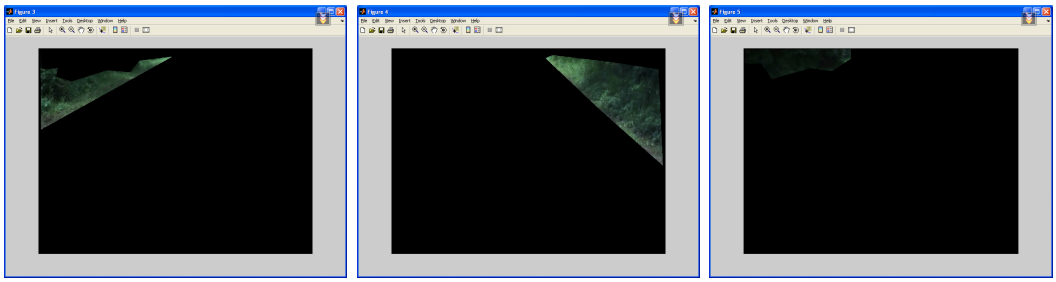
To overcome this problem, we observe that the ambiguous region can be

(a) Original image (b) Misclassification against dark areas (c) Improved classification result

Figure 5.12: Classification against dark areas.



(a) Original road image (b) Road region (blue)



(c) Vegetation region (cyan) (d) Vegetation region (red) (e) Upper left dark vegetation region (pink)

Figure 5.13: A typical road image and its segments. Manually segmented for plotting pixel distribution.

defined as a small cubic region spanning (50,50,50) to the origin (0,0,0). In particular, color pixels from dark areas (pink) are highly clustered into a small, dense cluster near the origin (0,0,0) and within that cubic region. It is also observed that only a small tip of the ellipsoidal road clusters are within that ambiguous region. Since it is assumed that road color pixels follow a Gaussian distribution, that means only a very small percentage of road pixels are within ambiguous region. Furthermore, even shadows on the road usually have higher brightness color, with minimum brightness above 50. Therefore, to reduce misclassification rate from very dark areas, those pixels with brightness values less than $B_T = 50$ are classified as non-road. The classification rule is, therefore, modified as shown in Algorithm 3. Figure 5.12(c) shows that this simple method can improve classification output significantly.

5.3.4 Post-processing

After classification, the classified image would still have outlier pixels. Some pixels on the road are classified as non-road, corresponding to small leaves, stones whereas some off-road pixels are classified as road, coming from similar-colored roadside buildings.

To remove impurities on the road region, we perform a morphological closing operation on the image. From the binary classified image, dilation and erosion operation were performed in sequence to join small holes within the images. This is based on the assumption that if there is a big patch of non-drivable section within the region of the image, there cannot be a few spots of drivable section within this region.

For removing false positives from roadside buildings, we perform a flood-fill operation to remove any "road" components not connected to the learning region. This improves the classified result because very often, background objects outside the drivable region that has the same color will be classified as the drivable region. The flood-fill will allow the most probable drivable region to be picked up. The choice of the seed for flood-fill is important because if the point chosen is wrong (e.g. a non-drivable region), then the wrong component will be picked up as drivable. To address this, several flood-fill operations are performed

Algorithm 3 Road classification algorithm

Require: Road color model with N Gaussians, each Gaussian is characterized by a mean vector μ_i, a covariance matrix Σ_i, and a mass number m_i.

Require: the largest mass number $m_{max} = \max_{i=1\to N}(m_i)$, minimum fraction $f_{classify}$, road-nonroad threshold $d_{classify}$, brightness threshold B_T.

1: **for** each image pixel $\mathbf{p}$ with (R_p, G_p, B_p) color values **do**

2: Compute brightness value and intrinsic values:

$$B_p = \frac{R_p + G_p + B_p}{3},$$

$$\mathbf{p} = \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix} = \begin{bmatrix} \log \frac{R_p}{B_p} \cos\alpha + \log \frac{G_p}{B_p} \sin\alpha \\ \log \frac{R_p}{G_p} \cos\beta + \log \frac{B_p}{G_p} \sin\beta \end{bmatrix}.$$

3: **if** $B_p < B_T$ **then**

4: $p =$ non-road

5: **else**

6: **for** $i = 1$ to N **do**

7: Select a Gaussian from N Gaussians of the current road color model.

8: **if** $m_i > f_{classify} \times m_{max}$ **then**

9: Compute distance from pixel to the current Gaussian.

$$d(\mathbf{p}, \mu_i) = ((\mathbf{p} - \mu_i)^T \Sigma_i^{-1} (\mathbf{p} - \mu_i))$$

10: **else**

11: The current Gaussian is insignificant and not considered.

$$d(\mathbf{p}, \mu_i) = \infty$$

12: **end if**

13: **end for**

14: Find the minimum distance from pixel to any Gaussian in the color model.

$$d_{min} = \min_{i=1\to N}(d_i)$$

15: **if** $d_{min} < d_{classify}$ **then**

16: $p =$ road

17: **else**

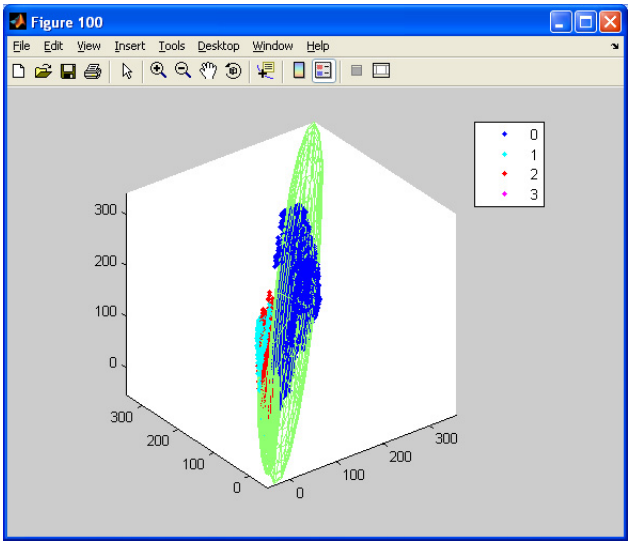
18: $p =$ non-road

19: **end if**

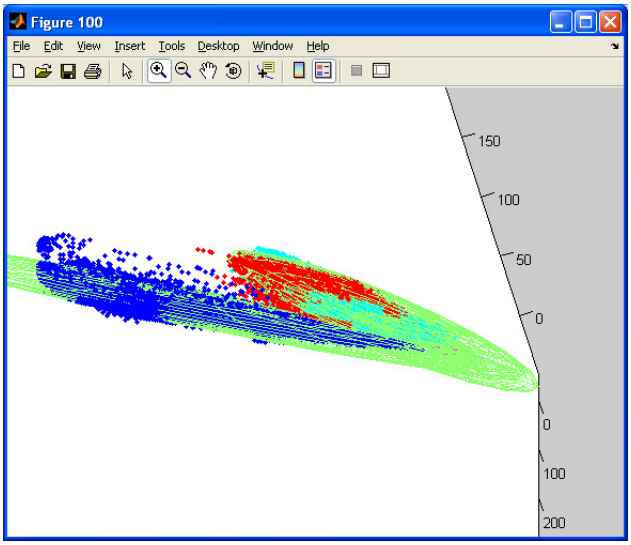
20: **end if**

21: **end for**

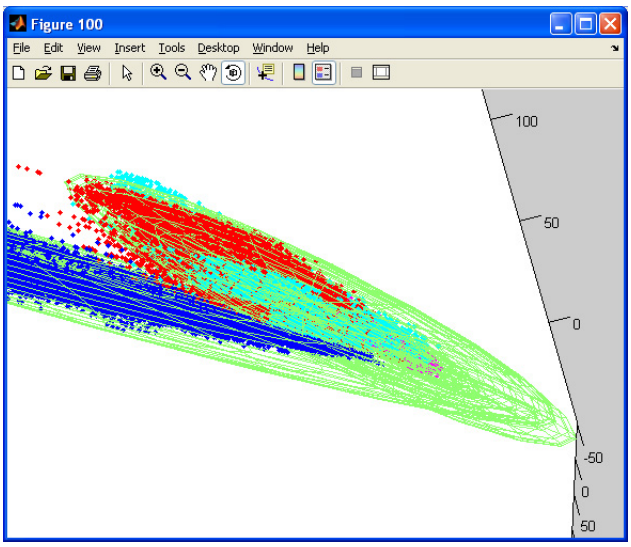
22: Return classified image.



(a) Color pixel distributions



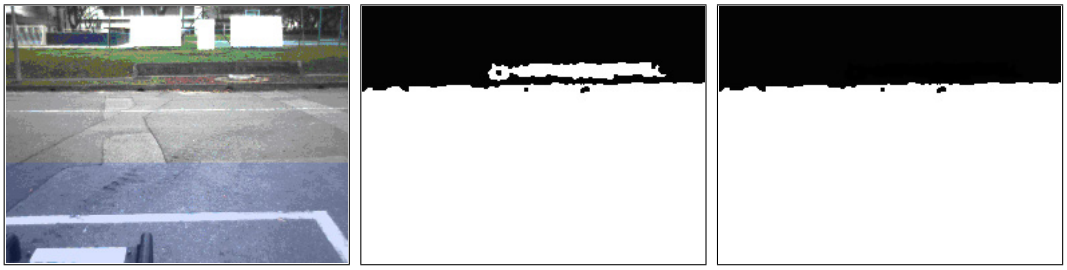
(b) Zoom in at (0,0,0)



(c) Further zoom in

Figure 5.14: Distribution of color pixels in RGB color space, from image in Figure 5.13. Road (blue), vegetation (cyan, red), dark vegetation (pink).

with seeds that vary across the width from the front of the vehicle (blue region in Figure 5.15(a)) and inside the training region. The seed that provides the maximum component size detected from the flood-fill operation is chosen. In our implementation, three flood-fill operations at random seeds are performed.



(a) Original road image (b) Classified road image (c) After flood-fill operation

Figure 5.15: Flood-fill operation.

Chapter 6

Results and Discussion

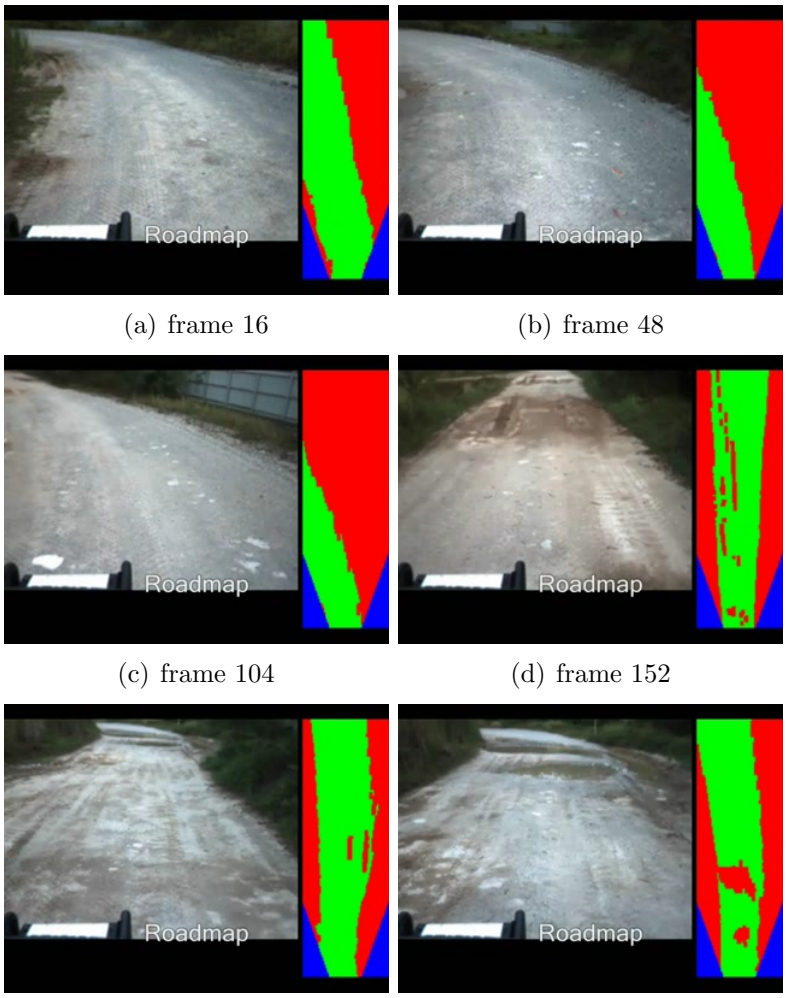
The above algorithm was extensively tested on several datasets that were collected real-time on a moving vehicle. Each dataset has more than 800 images, and they are collected on different driving terrains such as semi-structured rural roads, urban roads, and highways.

In our experiment, the algorithm is coded in C++ using the Open Source Computer Vision Library (OpenCV) [43]. The following parameters are used in the final version: $N = 10$, $c_k = 3$, $d_{similar} = d_{merge} = 1$ (Subsection 5.3.1), $d_{classify} = 3$, $f_{classify} = 0.3$ (Subsection 5.3.3). For each cycle (a pair of input stereo images), when running on a 1.86GHz Dell computer, the stereo processing step requires 0.25 second on average while the color-based learning and road extraction steps need 0.15 second. The color conversion is fast and takes insignificant computation time. In the worst stereo cases (see Chapter 4), the stereo processing step takes less than 0.5 second but those cases are rare.

The first section of the chapter presents the overall performance of the system in extracting road regions from the original color image and generating the top-view road-map. Next, the performances of individual components, namely the short-range stereo and the long-range road extraction, are presented. The performance of the long-range road extraction are analyzed quantitatively. We also compare our adaptive model number approach against the fixed model number approach [7], and demonstrate how our method works in the presence of shadows. Finally, the limitations of the current approach are discussed.

6.1 Overall performance

In Figures 6.1 and 6.2, we show the experimental results on a road section to demonstrate that our visual system is completely capable of extracting the road from a color image and transform it to a top-view grid map for navigation purposes. Although the top-view grid map is commonly used in navigation planning, it is not the final and optimal choice for our road-following application. Therefore, we will not go into detail about performance on road map outputs. Rather, we will discuss the classification performance of each component of our visual system.



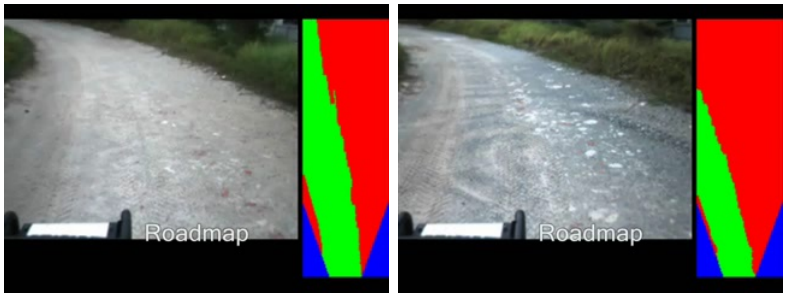
(a) frame 16 (b) frame 48

(c) frame 104 (d) frame 152

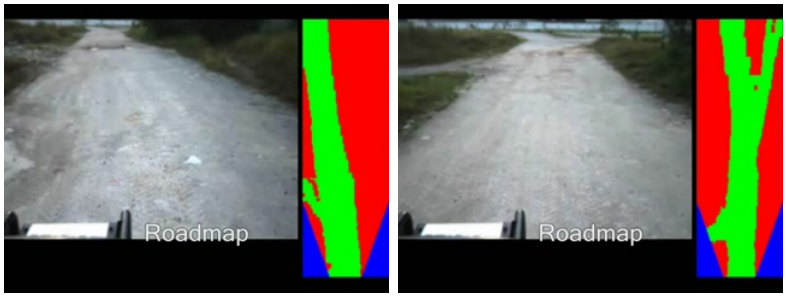
(e) frame 216 (f) frame 240

Figure 6.1: Top-view road map outputs and corresponding original images from a road image sequence. Road (green), non-road (red) and outside field of view (blue).

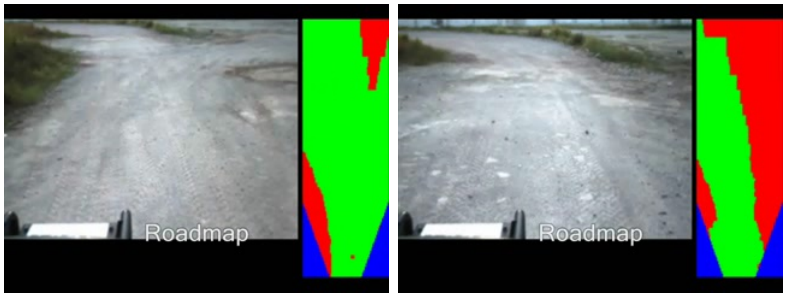
(a) frame 320 (b) frame 384



(c) frame 416 (d) frame 480



(e) frame 536 (f) frame 592



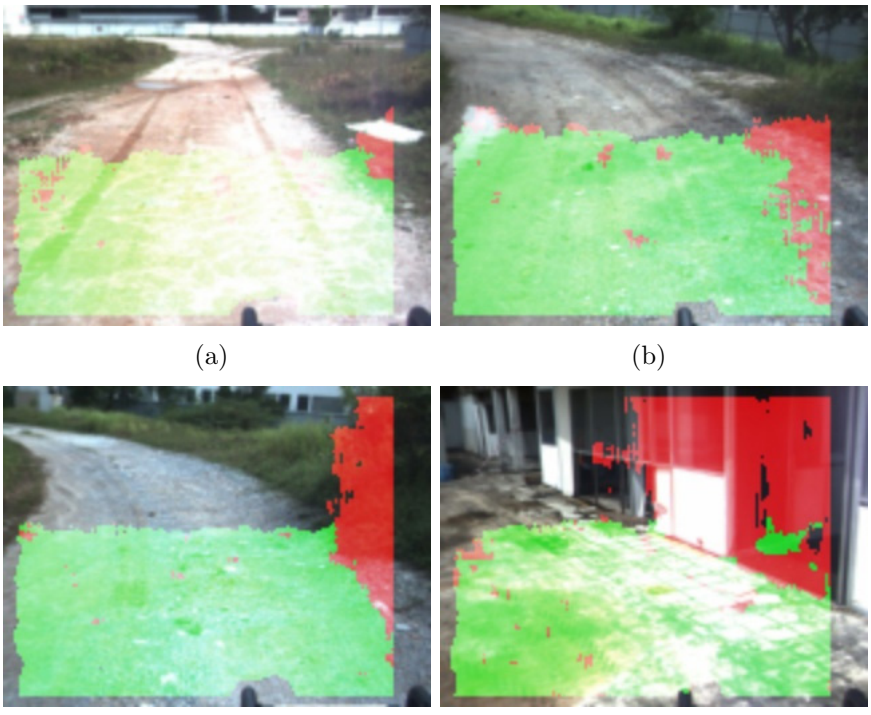
(g) frame 656 (h) frame 680

(i) frame 752 (j) frame 792

Figure 6.2: Top-view road map outputs and corresponding original images (cont.). Road (green), non-road (red) and outside field of view (blue).

6.2 Stereo-based obstacle detection

Some selected stereo results are presented in Figure 6.3. In general, the stereo module is able to detect obstacles within the 10-meter range, especially large obstacles that are dangerous to vehicle navigation. However, the stereo occasionally misses some obstacles with little or no features, such as plain white walls and homogenous surfaces. The detection of those obstacles is impossible for stereo algorithms as correspondence matching could be highly erroneous.



(a) (b)

(c) (d)

Figure 6.3: Some results of stereo-based obstacle detection. Obstacle regions are tinted red while ground regions are tinted green.

It is difficult to quantitatively analyze the performance of the short-range stereo module since it is difficult to collect 3D ground truth from a moving vehicle.

6.3 Adaptive number of models

Our algorithm performs well in different environments, as shown in Figure 6.4. It works satisfactorily on semi-structured rural roads, urban roads, and highways

although they are very different in texture and color.

Figure 6.4: Performance against different roads. Extracted road regions are tinted red.

In previous approaches that use color cues to extract roads, the number of Gaussian models are fixed [6] [7] [34]. In [7], the parameter values (k',N) of the Gaussian models that are trained and kept in memory are decided off-line. Setting k' too high would lead to overtraining issues while setting it too low would lead to erroneous classification. In Figure 6.5, we compared our adaptive model number approach against the fixed model number approach similar to [7] in the new color space. The results illustrate the advantage of training a flexible number of models.

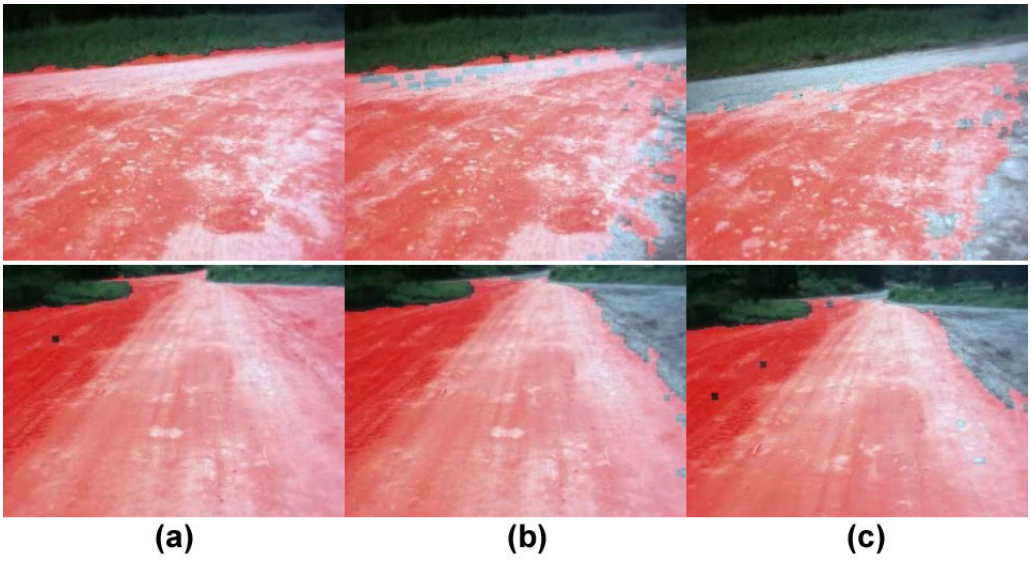


Figure 6.5: Comparison of performance against a rural road section: (a) Adaptive number. (b) $k' = 3$, $N = 10$. (c) $k' = 1$, $N = 4$.

6.4 Shadow-invariance

Figure 6.8 shows the outputs of road classification in RGB color space using the method discussed in [14] on a sequence of road images. At the start of the sequence, the upper part of the road was shadowed by roadside trees. Initially, in the first few frames, as there were no RGB color models available in memory, the shadows were classified as non-road. However, after a few frames, the algorithm gradually learned the color of the road and shadows and the road region was extracted correctly. Figure 6.7 shows the outputs of road classification using the current approach on the same shady road section. We note that even in frame 1, the shaded area at the far range posed no problem. Our algorithm learned the intrinsic colors of the road from the near-range learning region, and found that the shaded area has similar intrinsic colors. In RGB color space, such far-range shaded areas would be a significant challenge; shadows cannot be effectively handled early since RGB samples for shadows must be collected first.

In previous techniques such as [6] [7] [34], the authors either ignored training with shadow pixels and considered it as non-road or could not keep a model for shadows. Therefore, strong shadows in the image would possibly give the false perception of a dead-end road, as shown in Figure 6.6(a). Figure 6.6 demonstrates the significance of intrinsic colors in shadow-invariant road extraction.

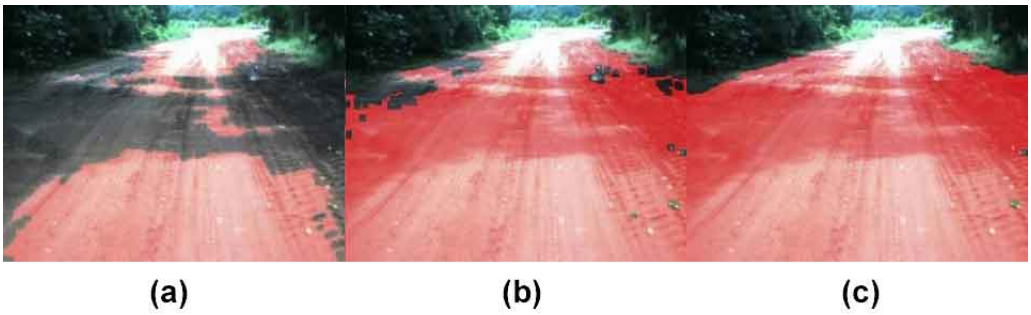


Figure 6.6: Comparison of classification methods against shadows: (a) RGB colors as in [8]. (b) Intrinsic colors, fixed model number. (c) Intrinsic colors, adaptive model number.

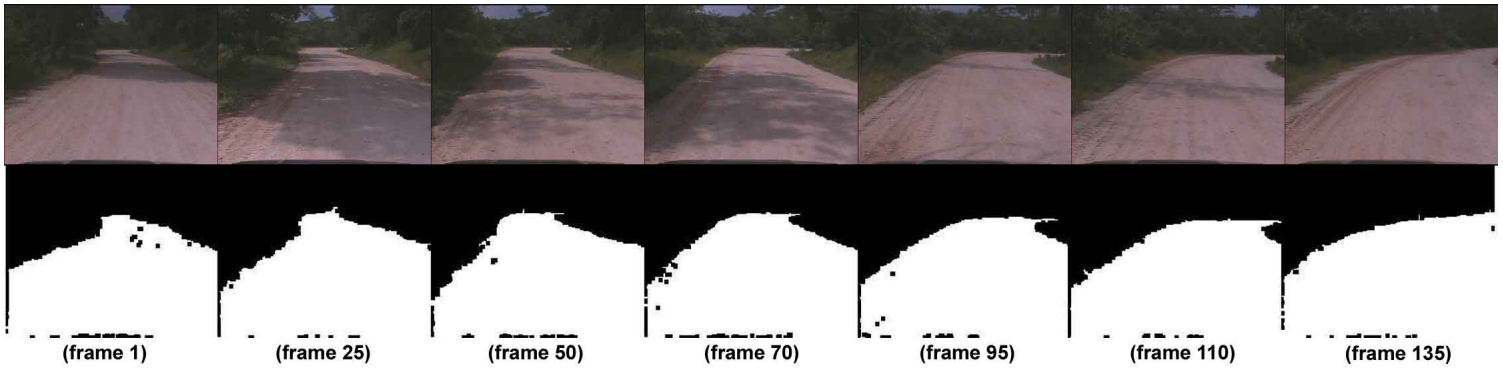


Figure 6.7: Performance against shadows in intrinsic color space on an image sequence.

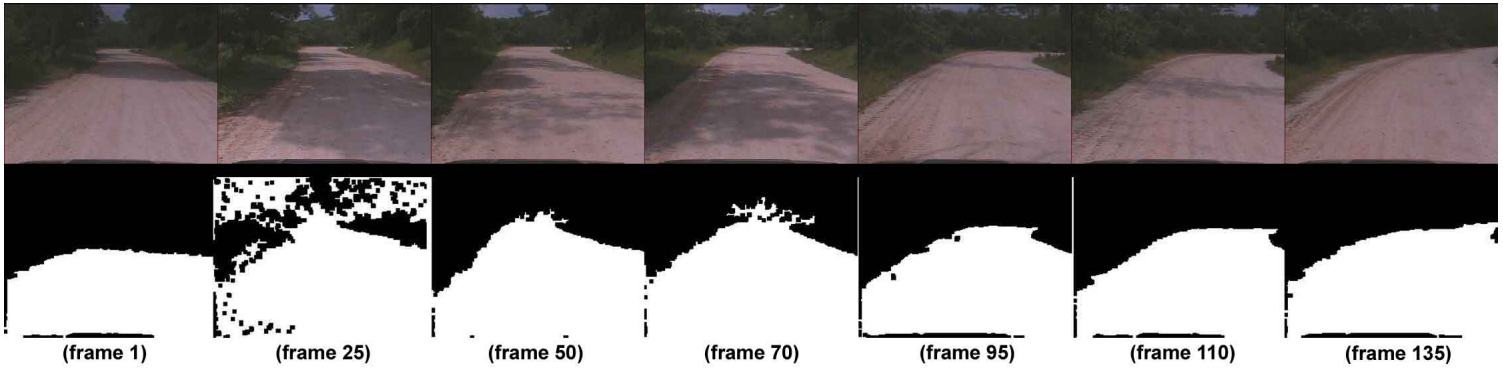


Figure 6.8: Performance against shadows in RGB color space [7][14].

6.5 Road extraction

In this section, the performance of the road extraction method is quantitatively analyzed to predict its reliability and usability. The performance is measured by two quantities, classification rate and usability rate. The classification rate is the average ratio of the number of pixels that are classified correctly to the number of pixels in the pre-defined ground truth. The usability rate is the average percentage of road maps that are usable for navigation purposes over the total output road maps.

The classification rate is commonly used as a general performance indicator for any classifier or classification method while usability rate is proposed to analyze the practical performance of road extraction methods in practical scenarios. These two measures are used together for quantitative analysis since, in many situations, the theoretical classification rate seems unsatisfactory while in fact, the usability rate is quite acceptable. This happens when a large number of pixels are misclassified in the images, compared to ground truth, but the extracted roads, especially the farther sections, are still correct and useful for navigation in terms of road shape and road orientation. In fact, the nearer road sections in the images close to the vehicle, are usually less significant for navigation planning but generally contribute most misclassified pixels since outliers such as stones, leaves on road are much more obvious at this range. Thus, the classification rate is generally a biased performance indicator for the purpose of our project.

6.5.1 Classification rate

To compute classification rate, each pixel in the classified image is compared one by one to a pre-defined ground truth. If the pixel in the classified image and the corresponding one in the ground truth are identical, i.e., road-road and nonroad-nonroad, we have a true road or a true non-road pair, respectively. When a pixel is classified as road in the image while it is non-road in the ground truth, we have a false road pair. Otherwise, when it is classified as non-road but it is actually non-road in the ground truth, the pair is a false non-road. The number

Table 6.1: Comparison of performance

	True road	True non-road	False road	False non-road
Dahlkamp	64.36%	21.93%	0.55%	13.16%
Our method	74.89%	21.10%	1.38%	2.63%

of pairs is counted for each type and divided by the total number of pixels in the image to obtain the percentage number. The process is repeated over a number of road scenes. The classification rate of the color classifier over the 8-km log file is shown in Table 6.1.

(a) Original RGB image (b) Classified result (c) Pre-defined ground truth

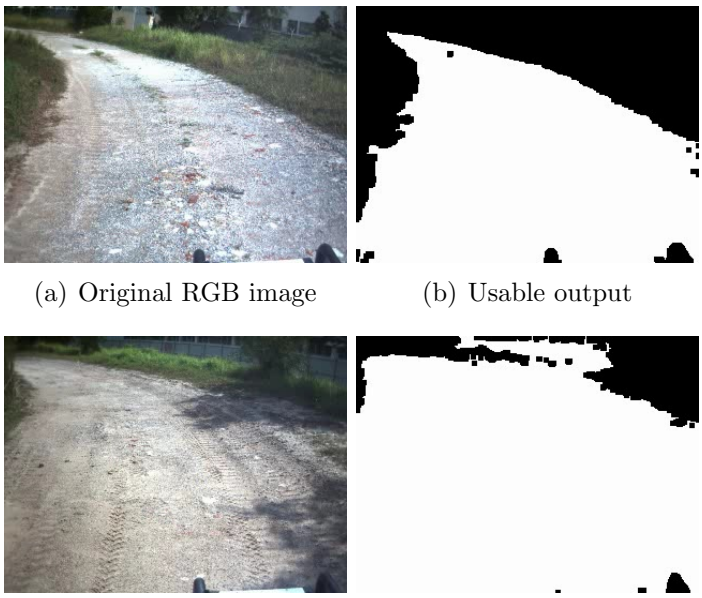
Figure 6.9: Original image, classified result, and pre-defined ground truth.

It can be observed that a large number of pixels that are falsely classified as non-road by Dahlkamp's method [7] are correctly classified as road by our method. Those pixels are mainly from shadow areas. In our method, the ratio of true-road to false-road pixels is relatively large, which is positive as it is undesirable for the vehicle to perceive an area as drivable while actually not and run into it. The percentage of false non-road is relatively higher as there are small parts on the road such as stones and small puddles that have different colors. These areas would be classified as non-road while in the ground truth, it is defined as road, leading to higher false non-road percentage. In addition, the percentage of false road is slightly increased as some image areas with dark colors similar to shadows on the road are misclassified as drivable. These areas usually corresponds to areas that are in shade at the far distance.

6.5.2 Usability rate

The classification rate only reflects partially the performance of the color classifier. To predict its reliability and usability for our project, usability is used to analyze the color classifier's performance. The classified output will be analyzed overall to see if it is useful for navigation. The average percentage of usable

outputs over the total outputs is the usability rate. In a "usable" output, it is not necessary that all the pixels are classified correctly.



(a) Original RGB image (b) Usable output

(c) Original RGB image (d) Non-usable output

Figure 6.10: Example of usable output (top) and non-usable output (bottom).

As shown in Figure 6.10, the top output is useful for navigation purpose although with some misclassified pixels as the road shape and road orientation is still maintained. On the other hand, in the bottom output, the white fence is misclassified as drivable which is dangerous for navigation. Therefore, this output is not usable.

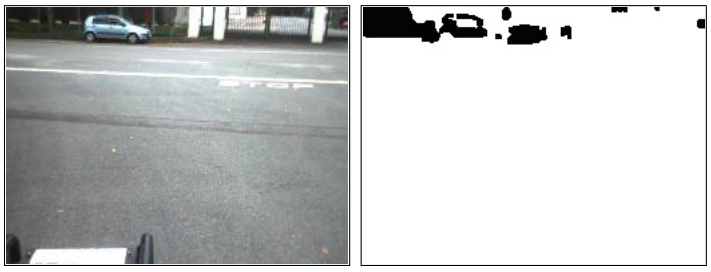
To compute the usability rate of the long-range road extraction module, the outputs of an 8-km run are generated and visually inspected. Since it is time-consuming to inspect every single frame, instead, random frame numbers from 1-5000 (approximate frame count of that 8-km sequence) are generated. In total, there are 250 frames inspected. Out of them, 232 outputs are rated as useful for navigation purpose. Hence, the usability rate can be estimated as 92.8%. Experiments with other road sections also give similar results, with computed usability rates above 90%.

6.6 Limitations

The current classifier strictly uses color information to classify the drivable and non-drivable portions of the road. This performs reasonably well on jungle track,

off-road environment. However, in an urban terrain, the current classifier faces a limitation due to the presence of rich color information in the environment. Because it uses color to find the road area, objects that are outside stereo range with similar R, G, B colors with road elements are misclassified. For example, the classifier will classify the white building besides the road as drivable because the lane markings on the road are white (Figure 6.11). It is impossible to resolve this issue by simply performing component detection. Rather the entire scene has to be analyzed so that the segmented result can be filtered. The solution to this will require using extra and complex information beyond color information to fine-tune the results.

The current approach uses intrinsic color as the illumination-invariant feature to deal with shadows. Although intrinsic color generally reflects correctly the road surface's intrinsic reflectance, it might fail when one of its assumptions or approximations is not valid. In particular, shadowed areas can receive significant illumination from reflected light from adjacent sunlit areas; and such inter-reflections are not modelled. Therefore, classification results are often noisy at the boundaries of shadows, and performance might be degraded significantly on road sections with intermixing shadows and lighted areas, such as those caused by sparse foliage of road-side trees. Severe over-exposure, caused by the camera pointing in the sun's direction, also affects intrinsic color value's consistency and classification performance.



(a) Original road image (b) Classified road image

Figure 6.11: Erroneous classified result for urban driving environment.

Chapter 7

Conclusion and Future Work

In this thesis, we have presented a vision-based system design and a new color space for robust road extraction in dynamic lighting conditions. These techniques have extended the capability of a camera sensor system for an autonomous vehicle or a driver-assistance application. The system consists of a pair of stereo cameras. The color information of the road at the near-range are collected based on stereo processing. The color models for the road are constructed and updated, in a new color space. The new color space is designed such that it represents the intrinsic reflectance information of the road surface and is independent of light sources. The algorithm aims to be adaptive in different environments by having a flexible number of color models constructed from sample pixels. Experimental results show that the proposed algorithm is able to handle shadows and perform adaptively for different driving environments.

The system presented in this thesis was successfully deployed in several real-time vehicle runs. However, several improvements can be made to increase the system robustness and usefulness.

The current long-range road extraction module performs learning and classification in intrinsic colors. Although this approach generally gives good performance, the intrinsic colors can be unstable and inconsistent when the color space assumptions and approximations become invalid. In particular, we assume that the camera's image sensors have narrow-banded spectral sensitivity functions such that they can be approximated by Dirac delta functions. However, the Bumblebee2's spectral sensitivity functions are not very narrow-banded, as

shown in Figure 2.4. This can be improved by making the spectral functions narrower, through spectral sharpening processes. Furthermore, the current road extraction algorithm is executed on individual images and purely based on color information. In the future, other image features as well as video tracking algorithms can be explored to improve the robustness since road extraction is mostly performed on consecutive road image sequences.

Besides the long-range module, the short-range stereo module can also be improved. Although the current range of 10 meters is adequate for sample collection and obstacle detection, the range could be further extended. This would allow the vehicle to achieve higher navigation speed as faster-moving vehicles would need more reaction time and distance to stop safely or evade obstacles. Better obstacle coverage would allow more efficient navigation planning. In conjunction with extended range, the ground estimation algorithm can also be improved. For a stereo range of about 20 meters or more, it would be too simplistic to assume that the ground is planar, as in the current ground estimation algorithm. In addition, another possible option for improvement is that once the ground surface can be accurately estimated, it can be used to perform homography projection from the classified road image into the top-view grid map.

Bibliography

[1] M. Agrawal, K. Konolige, and R. Bolles. Localization and mapping for autonomous navigation in outdoor terrains: A stereo vision approach. In *Proc. IEEE Workshop in Applied Computer Vision (WACV)*, 2007.

[2] H. Barrow, J. Tenenbaum, A. I. Group, and S. R. Institute. *Recovering intrinsic scene characteristics from images*. SRI International, 1977.

[3] J. Bouguet. Camera Calibration Toolbox for Matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

[4] P. Chaturvedi, A. Malcolm, and J. Ibanez-Guzman. Real-Time Road Following in Natural Terrain. In *Proceedings of 2004 IEEE Conf. on Cybernetics and Intelligent Systems*, volume 2, 2004.

[5] P. Chaturvedi, E. Sung, A. Malcolm, and J. Ibanez-Guzman. Real-time identification of driveable areas in a semistructured terrain for an autonomous ground vehicle. In *Proceedings of SPIE*, volume 4364, page 302, 2001.

[6] J. Crisman and C. Thorpe. SCARF: a color vision system that tracks roads and intersections. *IEEE Transactions on Robotics and Automation*, 9(1):49–58, 1993.

[7] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski. Self-supervised monocular road detection in desert terrain. In *Proc. of Robotics: Science and Systems (RSS)*, 2006.

[8] DARPA Administration. Grand Challenge 2004. `http://www.darpa.mil/grandchallenge04/`.

[9] DARPA Administration. Grand Challenge 2005. `http://www.darpa.mil/grandchallenge05/`.

[10] DARPA Administration. Grand Challenge 2007. `http://www.darpa.mil/grandchallenge/index.asp`.

[11] DARPA Administration. Learning Applied to Ground Robots (LAGR). `http://www.darpa.mil/IPTO/programs/lagr/lagr.asp`.

[12] E. Dickmanns and V. Graefe. Dynamic monocular machine vision. *Machine vision and Applications*, 1(4):223–240, 1988.

[13] T. Dong-Si, D. Guo, C. Yan, and S. Ong. Extraction of shady roads using intrinsic colors on stereo camera. In *IEEE International Conference on Systems, Man, and Cybernetics, 2008. SMC 2008*, pages 341–346, 2008.

[14] T. Dong-Si, D. Guo, C. Yan, and S. Ong. Robust extraction of shady roads for vision-based UGV navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008*, pages 3140–3145, 2008.

[15] G. Finlayson, M. Drew, and C. Lu. Intrinsic images by entropy minimization. *Lecture Notes in Computer Science*, pages 582–595, 2004.

[16] G. Finlayson, S. Hordley, and M. Drew. Removing shadows from images. *Lecture Notes in Computer Science*, pages 823–836, 2002.

[17] G. Finlayson and G. Schaefer. Hue that is invariant to brightness and gamma. In *Proc. British Machine Vision Conference*, pages 303–312, 2000.

[18] G. Finlayson, B. Schiele, and J. Crowley. Comprehensive colour image normalization. *Lecture Notes in Computer Science*, 1406(475-490):1406, 1998.

[19] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. 1981.

[20] R. Ghurchian, S. Hashino, and E. Nakano. A fast forest road segmentation for real-time robot self-navigation. In *2004 IEEE/RSJ International Con-*

ference on Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings, volume 1.

[21] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge Univ Pr, 2003.

[22] G. Healey and D. Slater. Global color constancy: recognition of objects by use of illumination-invariant properties of color distributions. *Journal of the Optical Society of America A*, 11(11):3003–3010, 1994.

[23] J. Jewett and R. Serway. *Physics for scientists and engineers with modern physics*. Wadsworth Publishing Co Inc, 2007.

[24] T. Jochem and D. Pomerleau. No Hand Across America. `http://www.cs.cmu.edu/afs/cs/usr/tjochem/www/nhaa/nhaa_home_page.html`.

[25] K. Konolige, M. Agrawal, R. Bolles, C. Cowan, M. Fischler, and B. Gerkey. Outdoor mapping and navigation using stereo vision. In *Proc. of the Intl. Symp. on Experimental Robotics (ISER)*. Springer, 2006.

[26] E. Land. Recent advances in Retinex theory. *Vision Research*, 26(1):7, 1986.

[27] E. Land and J. McCann. Lightness and retinex theory. *Journal of the Optical society of America*, 61(1):1–11, 1971.

[28] J. Lee and C. III. Road Following in an Unstructured Desert Environment Based on the EM (Expectation-Maximization) Algorithm. In *Proceedings of the International Conference on Control, Automation, and Systems*.

[29] X. Lin and S. Chen. Color image segmentation using modified HSI system for roadfollowing. In *1991 IEEE International Conference on Robotics and Automation, 1991. Proceedings.*, pages 1998–2003, 1991.

[30] Point Grey Research Inc. Bumblebee2 Getting started manual. `http://www.ptgrey.com/products/bumblebee2/index.asp`.

[31] D. Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979.

[32] D. Scott. *Multivariate density estimation: theory, practice, and visualization*. Wiley-Interscience, 1992.

[33] M. Tappen, W. Freeman, and E. Adelson. Recovering intrinsic images from a single image. *IEEE transactions on pattern analysis and machine intelligence*, 27(9):1459–1472, 2005.

[34] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer. Vision and navigation for the Carnegie-Mellon Navlab. *Annual Review of Computer Science*, 2(1):521–556, 1987.

[35] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al. Stanley: The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9), 2006.

[36] M. Turk, D. Morgenthaler, K. Gremban, and M. Marra. VITS-A vision system for autonomous land vehicle navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):342–361, 1988.

[37] I. Ulrich and I. Nourbakhsh. Appearance-based obstacle detection with monocular color vision. In *Proceedings of the National Conference on Artificial Intelligence*, pages 866–871. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2000.

[38] Universita degli Studi di Parma. The ARGO Project. `http://www.argo.ce.unipr.it/ARGO/english/`.

[39] P. Vora, J. Farrell, J. Tietz, and D. Brainard. Digital color cameras. 1. Response models. *Hewlett-Packard Laboratory Technical Report, No. HPL-97-53*, 1997.

[40] Wikipedia. D65. `http://en.wikipedia.org/wiki/D65`.

[41] Wikipedia. International commission on illumination. `http://en.wikipedia.org/wiki/International_Commission_on_Illumination`.

[42] Wikipedia. Standard illuminant. `http://en.wikipedia.org/wiki/Standard_illuminant`.

[43] Yahoo! Groups. OpenCV - Open Source Computer Vision Library Community. `http://tech.groups.yahoo.com/group/OpenCV/`.

[44] A. Youssif, A. Ghalwash, and A. Ghoneim. A Comparative Evaluation of Preprocessing Methods for Automatic Detection of Retinal Anatomy. In *Proc. 5th Int. Conf. Informatics Syst. (INFOS2007)*, pages 24–30, 2007.

Appendix A

Scott's rule for optimal histogram bin width

The histogram is an important statistical tool for displaying and summarizing data, providing an estimate of the true underlying probability density function. However, guidelines on how to construct a good histogram do not address some estimation issues and rely heavily on the investigator's intuition and past experience. In his paper [31] and subsequent book [32], Scott proposed a rule to compute the optimal bin width for histogram construction.

We consider a histogram calculated based on a set of data points $(x_1, x_2, \ldots, x_n)$, where n denotes the sample size. We must choose an optimal bin width h_n^* which determines the optimal smoothness of the histogram. We only consider histograms defined on an equally spaced mesh $t_{ni}; -\infty < i < \infty$ with bin width $h_n = t_{n(i+1)} - t_{ni}$. The subscript n is to emphasize the dependence of the mesh and bin width on the sample size.

For a fixed point x, the mean squared error (MSE) of a histogram estimate, $\hat{f}(x)$, of the true density value, $f(x)$, is defined by

$$\text{MSE}(x) = E\{\hat{f}(x) - f(x)\}^2 \tag{A.1}$$

The integrated mean square error represents a global error measure of a histogram estimate and is defined by

$$\text{IMSE}(x) = \int E\{\hat{f}(x) - f(x)\}^2 dx \tag{A.2}$$

Using some assumptions, Scott derives the following equations [31]:

$$\text{MSE}(x) = \frac{f(x)}{nh_n} + \frac{1}{4}h_n^2 f'(x)^2 + f'(x)^2\{x - t_n(x)\}^2 - h_n f'(x)^2\{x - t_n(x)\} + O(\frac{1}{n} + h_n^3) \tag{A.3}$$

$$\begin{aligned}\text{IMSE}(x) = &\frac{1}{nh_n} + \frac{1}{4}h_n^2 \int f'(x)^2 dx \\ &+ \int f'(x)^2\{x - t_n(x)\}^2 dx - h_n \int f'(x)^2\{x - t_n(x)\} dx + O(\frac{1}{n} + h_n^3)\end{aligned} \tag{A.4}$$

where $I_n(x)$ is the bin interval that contains a fixed point x as n varies and $t_n(x)$ denote the left-hand endpoint of $I_n(x)$. Scott shows that the equation A.4 can be further simplified:

$$\begin{aligned}\text{IMSE}(x) = &\frac{1}{nh_n} + \frac{1}{4}h_n^2 \int f'(x)^2 dx \\ &+ \underbrace{\int f'(x)^2\{x - t_n(x)\}^2 dx}_{\frac{1}{3}h_n^2 \int f'(x)^2 dx + O(h_n^3)} - \underbrace{h_n \int f'(x)^2\{x - t_n(x)\} dx}_{\frac{1}{2}h_n^2 \int f'(x)^2 dx + O(h_n^3)} + O(\frac{1}{n} + h_n^3)\end{aligned} \tag{A.5}$$

Therefore

$$\text{IMSE}(x) = \frac{1}{nh_n} + \frac{1}{12}h_n^2 \int_{-\infty}^{\infty} f'(x)^2 dx + O(\frac{1}{n} + h_n^3) \tag{A.6}$$

Minimizing IMSE in A.6, we obtain

$$h_n^* = \left(\frac{6}{\int_{-\infty}^{\infty} f'(x)^2 dx}\right)^{\frac{1}{3}} n^{-\frac{1}{3}} \tag{A.7}$$

which is the optimal choice for h_n.

For Gaussian sample data, i.e., $f(x)$ is Gaussian function, we have

$$\int_{-\infty}^{\infty} f'(x)^2 dx = \frac{1}{4\sigma^3\sqrt{\pi}} \tag{A.8}$$

$$h_n^* = (\frac{24\sigma^3\sqrt{\pi}}{n})^{1/3} \approx 3.49\sigma n^{-1/3} \tag{A.9}$$

Scott in [32] proposed the sample standard deviation s as an estimate of σ, resulting in the following Scott's rule:

$$h = 3.49\text{std}(\zeta)N^{-1/3}. \tag{A.10}$$

Appendix B

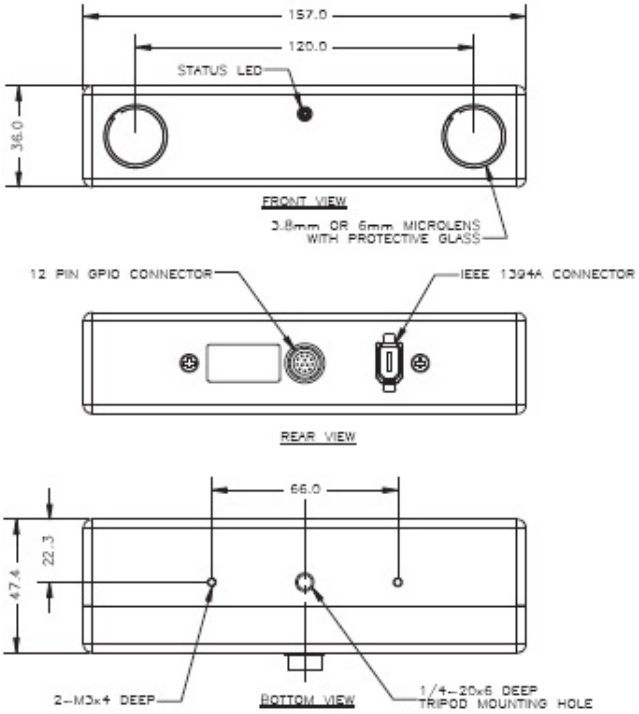
Bumblebee2's technical specifications

Camera Specifications

Specification	Low-Res (640x480)	High-Res (1024x768)
Imaging Sensor	Two Sony 1/3" progressive scan CCD	
	ICX424 (648x488 max pixels)	ICX204 (1024x768 max pixels)
	7.4µm square pixels	4.65µm square pixels
Baseline	12cm	
Lens Focal Length	2.5mm with 100° HFOV or 3.8mm with 70° HFOV or 6mm with 50° HFOV	
A/D Converter	Analog Devices 12-bit analog-to-digital converter	
Video Data Output	8, 16 and 24-bit digital data (see *Supported Data Formats* below)	
Frame Rates	48, 30, 15, 7.5, 3.75, 1.875 FPS	18, 15, 7.5, 3.75, 1.875 FPS
Interfaces	6-pin IEEE-1394 for camera control and video data transmission 4 general-purpose digital input/output (GPIO) pins.	
Voltage Requirements	8-30V	
Power Consumption	Less than 3W	
Gain	Automatic/Manual/One-Push Gain modes	
	0dB to 24dB	0dB to 24dB
Shutter	Automatic/Manual/One-Push Shutter modes	
	0.01ms to 66.63ms @15 FPS	0.01ms to 66.63ms @15 FPS
	Extended Shutter modes	
	0.01ms to 7900ms @ 15 FPS	0.01ms to 5200ms @ 15 FPS
Gamma	0.50 to 4.00	
Trigger Modes	DCAM v1.31 Trigger Modes 0, 1, 3, and 14	
Signal To Noise Ratio	Greater than 60dB at 0dB gain	
Dimensions	157mm x 36mm x 47.4mm	
Mass	342 grams	
Camera Specification	IIDC 1394-based Digital Camera Specification v1.31	
Emissions Compliance	Complies with CE rules and Part 15 Class A of FCC Rules	
Operating Temperature	Commercial grade electronics rated from 0° to 45°C	
Storage Temperature	-30° to 60°C	

Figure B.1: Bumblebee2 camera specifications. Retrieved from [30].

Physical Dimensions



Camera Features

Image Acquisition

Feature	Description
Automatic Synchronization	Multiple Bumblebee2's on the same 1394 bus automatically sync
Fast Frame Rates	Faster standard frame rates
Multiple Trigger Modes	Bulb-trigger mode, overlapped trigger/transfer
Color Conversion	On-camera conversion to YUV411, YUV422 and RGB formats
Image Processing	On-camera control of sharpness, hue, saturation, gamma, LUT
Embedded Image Info	Pixels contain frame-specific info (e.g. shutter, 1394 cycle time)

Camera and Device Control

Feature	Description
Frame Rate Control	Fine-tune frame rates for video conversion (e.g. PAL @ 24 FPS)
Strobe Output	Increased drive strength, configurable strobe pattern output
RS-232 Serial Port	Provides serial communication via GPIO TTL digital logic levels
Memory Channels	Non-volatile storage of camera default power-up settings
Temperature Sensor	Reports the temperature near the imaging sensor
Camera Upgrades	Firmware upgradeable in field via IEEE-1394 interface.

Calibration and Mechanics

Feature	Description
Lens System	High quality microlenses protected by removeable glass system
Accurate Pre-Calibration	For lens distortions and camera misalignments
Stereo Pair Alignment	Left and right images aligned to within 0.05[1] pixel RMS error
Calibration Retention	Minimizes loss of calibration due to shock and vibration

Figure B.2: Bumblebee2 camera specifications (cont.). Retrieved from [30].