Colour Vision and Hue for Autonomous Vehicle Guidance

Master’s Thesis project in Computer Vision at Linköping University by
Urban Bergquist

LiTH-ISY-EX-2091

Examiner: Ass. Prof. Hans Knutsson
Supervisor: Ass. Prof. Eric Sung

Linköping, December 1999
Library form
Abstract

We explore the use of colour for interpretation of unstructured off-road scenes. The aim is to extract driveable areas for use in an autonomous off-road vehicle in real-time. The terrain is an unstructured tropical jungle area with vegetation, water and red mud roads.

We show that hue is both robust to changing lighting conditions and an important feature for correctly interpreting this type of scene. We believe that our method also can be deployed in other types of terrain, with minor changes, as long as the terrain is coloured and well saturated.

Only 2D information is processed at the moment, but we aim at extending the method to also treat 3D information, by the use of stereo vision or motion.

Keywords

Autonomous off-road vehicle, visual guidance, real-time colour image segmentation, natural scenes, shading, hue.
Acknowledgements

I would most of all like to thank my supervisor associate professor Eric Sung at the School of Electrical and Electronic Engineering, Nanyang Technological University, for the project proposal and his assistance and guidance during the course of the project. I am also very grateful to Dr. Javier Ibanez-Guzman and Dr. Andrew A. Malcolm at Gintic Institute of Manufacturing Technology for their invaluable help and support.

Furthermore I would like to thank Dr. Fong Aik Meng and Robert Lee at Gintic for hosting me within their company during the project work.

I also thank associate professor Hans Knutsson at the Computer Vision Lab, Linkoping University, Sweden, for his support during the project as well as for having guided me into the field of neural networks.

Finally, I am tremendously thankful to all the friendly and interesting people I have met during my stay in Singapore. They made my stay an unforgettable experience.
Contents

1 INTRODUCTION ........................................................................................................1
  1.1 REPORT STRUCTURE .........................................................................................1
  1.2 PROJECT CONTEXT .........................................................................................1
  1.3 PROBLEM FORMULATION ...............................................................................1
     1.3.1 Objectives ...............................................................................................2
     1.3.2 Nature of challenge ...............................................................................3
     1.3.3 Scope .......................................................................................................3
     1.3.4 Constraints .............................................................................................3
     1.3.5 Definition of “driveable areas” ..........................................................4
     1.3.6 Formal statement of objectives ..........................................................4
  1.4 RESEARCH METHOD ......................................................................................5
  1.5 RELATED WORK ............................................................................................5
     1.5.1 Carnegie Mellon University ..................................................................5
     1.5.2 Other related work ...............................................................................7
     1.5.3 Colour image segmentation ...............................................................8
     1.5.4 Conclusion on related work ...............................................................8

2 BACKGROUND THEORY .....................................................................................9
  2.1 VISUAL GUIDANCE .......................................................................................9
  2.2 SEGMENTATION ..........................................................................................10
  2.3 BAYESIAN DECISION THEORY ....................................................................10
  2.4 SMOOTH THRESHOLDING .........................................................................12
  2.5 COLOUR ......................................................................................................14
     2.5.1 Representing colour ..........................................................................14
     2.5.2 Invariance to lighting conditions ..................................................19
  2.6 NEURAL NETWORKS ..................................................................................19
     2.6.1 General ...............................................................................................19
     2.6.2 Architecture .......................................................................................20
     2.6.3 Training .............................................................................................20

3 PROPOSED SYSTEM STRUCTURE ...................................................................23

ix
# 1 Introduction

## 1.1 Report structure

Initially we formulate the problem and state the objectives. We continue by presenting and analyzing related work to get ideas and learn from the experience of others and this leads to the proposal of a system structure for the application. Relevant theory is then presented and we present how to conduct the experiments. This is later used for exploring the possible solutions. The result of these experiments is summarized to arrive at a final method, whose performance is evaluated. After this follows a discussion on possible applications and suggestions for future improvements.

## 1.2 Project context

The study has been done as a 20 week project at Nanyang Technological University (NTU), in close cooperation with Gintic Institute of Manufacturing Technology, both located in Singapore. The project is a partial fulfillment of the requirements for a Master of Science in Engineering to be awarded by Linköping University, Sweden.

Gintic and NTU have previously conducted extensive research in the area of autonomous vehicle guidance and the results have been successful. This study is a part of their efforts to continue developing well-functioning autonomous vehicles for various outdoor applications.

## 1.3 Problem formulation

There are numerous occasions where autonomously operating vehicles are suitable, for instance in hostile environments as in mine fields, in volcanoes and even on other planets. In the future, autonomous vehicles
might even be more cost efficient than having human drivers, as they can run continuously for hours and hours and there are no salaries to pay. There is, however, a long way to go before machines can completely replace humans in this area.

The main problem is the vehicle’s ability to perceive and understand its surroundings. Humans rely heavily on vision for their understanding of what is around them and there seems to be no reason why vision would not work as well for machine scene interpretation. But giving this ability to machines is a far from trivial task. Images contain much information and the associated calculations are thus very demanding. It is therefore necessary to limit the information to treat. This can be done by extracting and processing only certain relevant image features while discarding the rest of the information.

Humans have colour vision because it gives us information of surface properties of objects around us. This information is vital for us to understand our world. We believe that colour would also be a good basis for successful machine scene interpretation.

1.3.1 Objectives

The aim of this project is to investigate the usability of colour information for guiding an autonomous vehicle in an unstructured tropical jungle terrain.

The aim is to propose a method that is as general as possible, without unnecessary limiting assumptions and ad hoc solutions. Such a general solution will still be useable with minor adjustments when the operating conditions change somewhat.

We will implement a real-time system that is based on colour to segment the scene into driveable and non-driveable areas. Different image features will be explored as well as different classification methods. The system input is an image sequence from a camera mounted on an off-road vehicle and the output is a map of currently driveable areas in front of the vehicle. The map will be used for path planning in a vision system for off-road autonomous navigation.

The terrain is a typical Singaporean off-road area with vegetation, water and red mud roads.
1.3.2 Nature of challenge

Changing lighting conditions often cause problems in computer vision and making the system robust to lighting will be a main issue. Furthermore, we are restricted by the demand for a real-time application, so to arrive at a suitable method we will have to compromise between simplicity, for speed, and a more thorough method, for robustness and reliability. The first challenge is to identify and extract image features that are relevant for identifying driveable areas in a scene, and that are preferably also invariant to lighting. Secondly, we need a method to interpret these features, i.e. the actual segmentation of the scene.

1.3.3 Scope

We will evaluate the usefulness of colour for scene interpretation, i.e. location of driveable and non-driveable areas, for the specific terrain given by the operating scenarios. Focus will be given to find areas that are definitely driveable, while some areas that are not for sure driveable will be left out. This approach is chosen for vehicle safety reasons.

The study will not cover stereo vision or other approaches for deriving 3D knowledge about the scene nor will it involve any attempts to perform obstacle detection or recognition. Considering that no 3D knowledge is available, no mapping from 2D image information onto a 3D world map will be carried out.

1.3.4 Constraints

The main constraint is the demand for a real-time system, i.e. the proposed algorithm must involve simple and efficient calculations.

Only 2D information is available and this naturally limits the possibility to correctly interpret the traversability of a terrain. Driveability will only be with respect to the type of ground, while dangerous slopes, holes and bumps will be left out without consideration.

The only available information is the output from a standard consumer grade colour CCD camera, for visible light.

The small set of available operating scenarios limit the possibility to evaluate the system for more general conditions. The evaluation will be subjective due to the unstructured nature of the problem, no absolute
Introduction

A correct answer exists for comparison and quantification of the performance.

1.3.5 Definition of “driveable areas”

The aim is to find “driveable areas” in front of a vehicle, but before we continue we need to define driveable, since it highly depends on the context.

In practice, driveability depends on the vehicle. A golf cart may get stuck in a puddle of muddy water, while even the golf cart itself may not stop a tank rushing forward. For this application we assume a vehicle which in off-road terrain performs somewhere in between the two, a vehicle that is reasonably suited for off-road operation.

In addition we add a more practical assumption – driveable areas are always made of red mud/soil. This implies that grass and vegetation will never be considered as driveable, even though it might very well be, with the appropriate vehicle. Apart from being a practical assumption this is also a matter of safety, since grass and vegetation may hide potential vehicle dangers, like holes.

1.3.6 Formal statement of objectives

To summarize, we intend to do the following:

Investigate if colour is useful for finding driveable areas (refer to 1.3.5 Definition of “driveable areas”) in the operating scenarios.

Show this usefulness by implementing a method with the following properties:

- Input: 2D colour image sequence from standard colour CCD camera.
- Output: Map of driveable areas in front of the vehicle including a confidence measure.
- Constraints: Valid for the available operating scenarios.
- Performance: Robustness and calculation speed are essential issues.
What we will not cover:

- 3D information.
- Object detection and/or recognition.

1.4 Research method

The study started with a bibliographical search for related work and relevant theory. We then defined what we wanted to do and how we wanted to achieve this. The possible solutions were explored by testing their performance on the operating scenarios and we finally arrived at a method, that was evaluated.

1.5 Related work

We present related work in the domains of autonomous vehicle guidance and colour image segmentation.

1.5.1 Carnegie Mellon University

The CMU in Pittsburgh, USA, have conducted much research in the area of autonomous outdoor operation of vehicles. Here follows a summary of their most important projects.

ALVINN

The ALVINN (Autonomous Land Vehicle In a Neural Network) project was a part of the NavLab project at Carnegie Mellon University, Pittsburgh, USA [Pomerleau 1993] [Pomerleau & Jochem 1]. By using connectionist techniques (Neural networks) to treat the image sequence from a camera mounted on a vehicle, they managed to make it drive autonomously on highways for impressive distances. In addition to being very reliable, the system is able to learn a new environment by just “watching” a human driver drive for about 5 minutes.

Several interesting approaches were introduced in the ALVINN project. A feed-forward fully connected neural network with a “retina” of 30x32 input units is used to output a steering direction from the inputted image sequence. Good network training is realized through some interesting
techniques to guarantee training set diversity, including selective buffering, digital transformation and addition of structured noise. Prior to the neural network, the input image is reduced to 30x32 blocks. Each block is represented by a combination of RGB and normalized RGB. In fact only the blue content of the image (apart from the normalizing) is used according to an ad hoc formula: \[ b = \frac{B}{255} + \frac{B}{R + G + B} \]

This formula was empirically found to suppress the effects of shadows.

Even though ALVINN is a very interesting system, it does not fulfill our needs. The operating scenarios are different since it operates preferably on structured paved road. There are always one correct steering direction and hence a reactive output is used while we want an output in the form of a map.

Ralph

RALPH [Pomerleau & Jochem 2] stands for Rapidly Adapting Lateral Position Handler and is a system for autonomous road-following. It has been shown to successfully drive on a wide variety of roads and under very different weather and lighting conditions. The system received much attention in 1995 during the No Hands Across America trip [Pomerleau & Jochem 3], when it drove 98.2% of the 2849 miles between Washington DC and San Diego, CA.

The method involves image shifting and column summation to indirectly search for linear patterns in the direction of the road. The obtained pattern is then examined by matching to find the current lateral position of the vehicle. The road curvature is another output from the system. The techniques used are amazingly simple but efficient. However they are best suited for the relatively structured terrain of man-made roads and not for operation in highly unstructured off-road terrain.

Ranger

RANGER (Real-time Autonomous Navigator with a Geometric Engine) [Kelly & Stentz] is a software control system for cross country autonomous vehicles.
The approach involves world map construction based on information from a laser range finder and a stereo perception system. The system output is the same as what we desire, a map, but colour is not used as the prime criterion for determining the traversability of the terrain.

Nomad

The Nomad [Whittaker et al.] has been developed by CMU mainly for space exploration, but also for operation in distant and difficult environments like Antarctic. The project went through extensive testing in 1997 during the so called Atacama Desert Trek [IMGNASA 1998], when the vehicle was teleoperated on a 200 km trip through the Atacama Desert in Chile under conditions analogous to those found on the surfaces of the Moon and Mars.

The robot has three different modes, autonomous operation and supervised and unsupervised tele-operation. The autonomous and supervised modes are realized through the use of three pairs of stereo cameras and a laser range finder, among other things.

The operating scenario is dramatically different from ours and this is naturally reflected in the system design. Colour is not a relevant feature for segmenting between traversable and non-traversable stones.

1.5.2 Other related work

Successful implementation of visual guidance for autonomous road vehicles has also been done at the University of the Federal Armed Forces Munich, Germany [UniBwM] [Gregor et al. 1997], which has been conducting research in this area since the mid 1980’s. Colour has not been a main issue, probably since they deal mostly with paved roads. Their systems have been put through extensive testing on the German public highways ‘Autobahn’ and the results are impressing.

For operation on structured man-made roads, a normal and usually acceptable assumption is a flat world. For this model, [Hermans 1999] shows the importance of correctly locating the horizon in the scene, i.e. indirectly determining the tilt of the vehicle. However this will probably be hard to implement successfully for off-road operating scenarios.
1.5.3 Colour image segmentation

Processing of colour images normally starts with a colour segmentation of the image to divide the image into different regions of interest. This is one of the toughest problems in computer vision and many different techniques have been studied during years of research.

[Guo & Xie 1998] suggest an approach based on RCE (Restricted Coulomb Energy) neural networks and clustering of colours in 3D colour space, which seems to give good results. This approach shows good results, but is not suitable for our application since we require fast real-time implementation and furthermore this method is designed for far more complex segmentation tasks than identification of red and green areas. A simpler approach will probably be more suitable.

Hue and saturation have previously been used in colour recognition tasks, for instance for high-speed automatic inspection in the food industry [Batchelor & Whelan]. The environment is in this type of applications naturally much more controlled than what we face in our application, especially the lighting conditions, but there are several similarities.

1.5.4 Conclusion on related work

None of the above mentioned systems are fully applicable in our case, since the operating scenarios and system demands differ significantly from ours. However different parts are interesting, for instance how they obtained an image representation that was robust to shading in the ALVINN project. The successful implementations of hue and saturation based approaches by Batchelor & Whelan also encourage us to continue exploring hue as a relevant feature for segmentation.

In addition, they all provided knowledge of problems that arise in visual guidance of autonomous vehicles, knowledge that would prove useful during the course of our study.
2 

Background theory

We are to implement visual guidance for an autonomous vehicle, based on colour vision. We need to make the machine “understand” its environment, but this task, which is more or less trivial to humans, is all but trivial to implement in machines and different approaches can be taken, as seen in the related work in section 1.5 Related work. Due to the complexity of the problem, the “understanding” is often limited to segmentation of the scene into disjoint areas of interest that present similar properties. This segmentation is based on some features which are found relevant for the current task. Careful selection of these features is critical for the success of the implementation.

2.1 Visual guidance

Visual guidance is a very appealing approach to vehicle guidance, but it is not at all a trivial one. These systems often incorporate advanced computer vision techniques, as briefly described above, but the demand for high-speed processing is often a limiting constraint. The output can be of two main types, either a reactive output [1,12], that directly gives an appropriate steering command, or an output in the form of information for map building [13]. The so created map is then fed into a path planning algorithm. 3D knowledge is often very important, especially for the latter type of system. This can be achieved by using either stereo vision or a separate range finder, like radar or perhaps preferably a laser range finder.

Feature extraction is usually the first step and an important part of the processing. The choice of feature depends on both the scenario and what kind of output we want from the system.
2.2 Segmentation

Segmentation in image analysis is the division of an image into disjoint areas of interest based on some chosen image features. A number of different features can be used for segmentation, like colour, orientation and local frequency.

The segmentation procedure normally consists of two distinct parts, the feature extraction and the actual segmentation [Granlund & Knutsson 1995]. In the feature extraction, differences in the properties in focus are mapped into differences in some well chosen representation. The segmentation part then extracts homogeneous regions by applying certain rules to this representation space. Discrimination is a natural part of the segmentation problem.

The discrimination is generally done by the application of a linear discriminant function, but nonlinear approaches also exist, for instance neural networks. Segmentation is then usually done by initial classification using thresholding of the discrimination output followed by the grouping of homogeneous regions.

2.3 Bayesian decision theory

It is an ubiquitous assumption to state that the groups subject to classification have a gaussian distribution, also referred to as normal distribution. The probability function of gaussian distributions can be written as

\[ f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \]

where \( \mu \) is the group mean and \( \sigma \) the standard deviation, which describes a function according to Figure 2.1.
The assumption of a gaussian distribution is often not too far from reality and the advantage is that we then can apply the thorough formal theory that exists on gaussian distributions. For instance we can apply Bayes decision theory [Yamany 1996] for the classification task.

Bayes decision theory is based on the assumption that all relevant probabilities are known. Bayes decision theory states that the sample \( x \), which is an outcome of a process \( X \), shall be classified as belonging to a group \( \omega_i \) if \( \omega_i \) maximizes \( P(\omega_i \mid x) \), i.e.

Decide \( \omega_i \) for \( \max \{ P(\omega_i \mid x) \} \) where \( P(\omega_i \mid x) \) can be found by using Bayes' theorem which states that

\[
P(\omega_i \mid x) = \frac{p(x \mid \omega_i)P(\omega_i)}{p(x)}
\]

where \( P(\omega_i) \) is the a priori probability, \( P(\omega_i \mid x) \) is the a posteriori probability, \( p(x \mid \omega_i) \) is the conditional density function and \( p(x) = \sum_i p(x \mid \omega_i)P(\omega_i) \) is the density function.

The principle of Bayes decision theory can be formulated as classify sample \( x \) as belonging to group \( \omega_i \) if, given that we have the outcome \( x \), it is most probable that \( x \) belongs to \( \omega_i \), which seems logical. This reasoning will result in a classifier with certain discrete thresholds that separate the feature space and indicate the group membership of the outcomes \( x \) in that feature space.

Figure 2.1. Two examples of normal distributions.
In our classification task we will apply Bayes decision theory in the sense that an area will be classified as driveable if it is most likely that it belongs to the group of driveable areas, based on our knowledge about the current statistics. However, we will also consider the probability with which it belongs to that group, by the use of a smooth threshold function in contrast to the discrete that is traditionally used with bayesian classification.

**2.4 Smooth thresholding**

For the smooth thresholding we chose a sigmoid function (tangent hyperbolicus). The function is often referred to as a "squashing function" for its ability to "squash" a variable ranging [-oo, +oo] to a range of [-1, +1]. For discriminating hue, this is however not the prime use, but we more value it for being an good smooth threshold function, as can be seen in Figure 2.2. At a certain threshold value the output will go more or less rapidly from one extreme to the other. There are two parameters that can be varied, the desired threshold and the smoothness of the function. Figure 2.2 shows example plots of the function for some different parameter values.

![Figure 2.2. The sigmoid function for varying threshold and smoothness parameters.](image)

The sigmoid function is calculated as
Background theory

\[
f(x) = \frac{e^{\frac{x-x_0}{s}} - 1}{e^{\frac{x-x_0}{s}} + 1} \quad \text{where } x_0 \text{ is the desired threshold and } s \text{ is the smoothness parameter.}
\]

When using a discrete thresholding for block averaging the only output will be to which group the sample belongs. There would be no indication of how far from the threshold it is located. The basic assumption for the following reasoning is that the threshold is the best way to separate the two groups in question and that the further away from the threshold we are, the more certain we are of a correct classification.

Probabilities can be used to justify the use of a smooth threshold. If we consider two groups, \( \omega_1 \) and \( \omega_2 \), with gaussian distribution and equal group a priori probabilities, we can plot their theoretical distributions \( p(x|\omega_i) \) as in Figure 2.3, left plot. The a posteriori probability of belonging to one group or the other is

\[
P(\omega_i|x) = \frac{p(x|\omega_i)}{p(x|\omega_i) + p(x|\omega_2)} \quad \text{(from Bayes' theorem)}
\]

which is illustrated in Figure 2.3, right hand side plot, for group 2. This probability is plotted along with a smooth threshold function, a sigmoid function with suitable parameters. The sigmoid captures the characteristics of the probability relatively well and we thus consider it justified to use it for indicating the probability of a correct classification.

![Theoretical distributions of the groups (left) and the classification probability compared with a sigmoid function.](image)

Figure 2.3. Theoretical distributions of the groups (left) and the classification probability compared with a sigmoid function. (These are in fact the theoretical distributions of the hue values of the pixels in the test set as determined in Table 4.1. The real distributions are shown in Figure 4.9.)
By outputting a continuous measure in the range [-1, 1] we get an estimation of how certain we can be of the classification made, in addition to the actual classification result which is a discrete group. An output of +0.5 would indicate that we have approximately 75% chance of having the group +1, even though we naturally can not expect a direct correspondence between the output and the true Bayesian probability without studying the statistics more thoroughly. But it is at least an indication. This indication is being used as one component for estimating the overall confidence in section 4.6 Confidence measure.

Smooth thresholding has previously been studied for use in different application areas of decision theory, for instance in the domain of medical informatics [Bergquist & Babic 1999].

### 2.5 Colour

We believe that colour is a highly relevant feature for segmentation between driveable and non-driveable areas in the operating scenario. However we need a representation of colour that highlights the differences between the different regions of interest and is invariant to things that are irrelevant for the segmentation process. A typical irrelevant feature is changing lighting conditions, since the lighting conditions have nothing to do with the properties of the terrain in front of the vehicle. Invariance to changing lighting conditions is therefore a much desired property.

#### 2.5.1 Representing colour

The human eye decodes the incident light using two types of receptors namely rods and cones. Rods determine the intensity of light while there are three types of cones. They are sensitive to red, green and blue wavelengths. Hence three different dimensions are enough to describe the human perception of colour. Colour spaces are the mathematical representations of the phenomenon we perceive as colour.
RGB

The RGB (Red, Green and Blue) colour space is the de facto standard representation in the world of computers and digital cameras and is therefore often a natural choice for colour representation. Its disadvantage is that it is hard to understand for a human observer, since it does not correspond to the way we commonly describe and analyze colours. It is hard to separate the properties of light we usually refer to when using RGB components. An important property like brightness is not separate from other important properties like “which colour”, but is to be found in all components. None of the components will hence be invariant to for instance changing lighting conditions.

Advantages: Standard. No calculations, since this is what you get from CCD-cameras.

Disadvantages: No component invariant to lighting. Hard to understand.

Normalized RGB

Normalized RGB has been introduced as an attempt to make the representation less sensitive to lighting effects. Normalized RGB is defined as:

\[
r = \frac{R}{R + G + B} \quad g = \frac{G}{R + G + B} \quad b = \frac{B}{R + G + B}
\]

This representation is often used in computer vision applications to limit the effects of changing lighting conditions, such as shading.

Advantages: Easy calculations. Fairly good results.

Disadvantages: Hard to understand.

Hue, saturation and brightness

An intuitively appealing representation that corresponds to the artists concepts of tint, shade, and tone, respectively. This is in fact more a family of colour spaces since there exists a wide variety of colour spaces based on the concept of tint, shade and tone, all with slightly different representations and nomenclature. Examples are the HSV, the HSL and the HSI, to name a few.
Hue, saturation and brightness have all been defined by the Commission Internationale de l’Eclairage (CIE) [CIE 1987]. Hue is defined as “the attribute of a visual sensation according to which an area appears to be similar to one of the perceived colours red, yellow, green and blue, or a combination of two of them”. Saturation is “the colourfulness of an area judged in proportion to its brightness” and brightness is “the attribute of a visual sensation according to which an area appears to emit more or less light”.

Or to put it in other words: Brightness corresponds to how light or dark a pixel appears. Saturation is how coloured a pixel is, i.e. how much it deviates from a tone of gray. Hue, finally, is which colour we have, i.e. red, yellow, green, cyan, blue, magenta or a combination of them. The colour space can also be illustrated by a “double cone” as shown in Figure 2.4. The left image shows the “double cone” and the right image illustrates hue and saturation on the hexagonal plane orthogonal to brightness.

The perception of brightness is in reality very complex [Poynton 1997] and a simpler representation is CIE luminance, denoted Y. Luminance is the radiant power weighted by a spectral sensitivity function that is characteristic for human vision.
An easy way to obtain HSY representation is by using a so-called colour difference representation, like \( Y'P'_R \) (the ' denotes a non-linear quantity) which is commonly used in television and video systems. When RGB is transformed to the \( Y'P'_R \) representation, \( Y' \) will contain the luminance and \( P'_R \) and \( P'_B \) the red and blue components, respectively, with the luminance taken away. This transformation is described in [Poynton 1997]. The hue is then computed as the angle between the orthogonal \( P'_B \) and \( P'_R \) components. The representation of hue, saturation and luminance, that we obtain this way, we refer to as HSY'.

\[
\begin{bmatrix}
 Y' \\
 P'_R \\
 P'_B 
\end{bmatrix} =
\begin{bmatrix}
 1 & 0 & 0 \\
 0 & 0.5 & 0 \\
 0 & 0.886 & 0.5 \\
\end{bmatrix}
\begin{bmatrix}
 Y'_{601} \\
 B' - Y'_{601} \\
 R' - Y'_{601} 
\end{bmatrix} =
\begin{bmatrix}
 0.299 & 0.587 & 0.114 \\
 -0.168736 & -0.331264 & 0.5 \\
 0.5 & -0.418688 & -0.081312 
\end{bmatrix}
\begin{bmatrix}
 R' \\
 G' \\
 B' 
\end{bmatrix}
\]

where \( R', G', B' \in [0,1] \)

\[
Y'_{601} \in [0,1] \quad P'_B, P'_R \in [-0.5, +0.5]
\]

\[
H = \begin{cases} 
\frac{1}{\pi} \arctan \left( \frac{P'_R}{P'_B} \right) - \frac{\pi}{2} & \text{for } P'_R > 0 \\
\frac{1}{\pi} \arctan \left( \frac{P'_R}{P'_B} \right) + \frac{\pi}{2} & \text{for } P'_R < 0 
\end{cases}
\]

For \( P'_R, P'_B = 0 \) special cases apply to obtain a continuous hue representation.

This method is slightly different from most other methods for obtaining hue, saturation and brightness. The major difference is that the hue-saturation plane will not be a hexagon as for most others, but a circle according to Figure 2.5.
Poynton argues that hue, saturation and brightness based representations should be abandoned, using the argument that it is not suitable for conveying accurate colour information, an often unsatisfactory definition of the brightness component and colour discontinuities due to poor algorithms. However, we still consider it a useful representation, since we do not depend on such an accurate colour conveyance between different devices. Furthermore, we have chosen a standard brightness component, the $Y'_{601}$ from the definition of video $Y'P_bP_r$, which does not possess the deficiencies usually found in more naive brightness representations. Finally, we compute hue in a way that gives a continuous hue.

There will however be one discontinuity in the hue representation, since the hue is an angular measure and it is impossible to have a continuous one-dimensional 1-to-1 representation of angular values. The fact that hue is not defined for tones of gray should be noted.

Advantages: Easy to understand. Better result?
Disadvantages: More complex calculations. Hue is undefined for gray.

Other colour representations

There are several more colour spaces that are suitable for different applications. Some of them focus on uniformity, i.e. having linear correspondence between the perceived difference between two colours.
and the distance separating them in the colour space. However, this often desired property is redundant for this application.

2.5.2 Invariance to lighting conditions

Treatment of colour images is considered by many as the most difficult task in computer vision. One big issue is how to achieve colour constancy, i.e. how to make the colour perception depend only on the actual colour of the object and not on the lighting. For our purposes, we will only deal with sun light and the problem then becomes more limited, but we still have the problem of shading which is complicated enough. In addition to changing the intensity, shadows cause a shift of colours towards blue as shown in [Pomerleau 1993], since shadows are only lit by the blue sky and not by the white sun beams like the rest of the scene. Another contributing factor is non-linearities in modern colour CCD cameras, at low intensities these tend to be more sensitive to blue light than to red and green. This is hence a very complex problem.

The ALVINN project [Pomerleau 1993] uses an ad hoc formula to solve the problem. Colour constancy is otherwise studied by several research groups world-wide, for instance at Simon Fraser University Canada [SFU], but the focus is often enhancement of picture quality and correction for non-white illumination of scenes and not shading.

2.6 Neural networks

We have considered the use of neural networks (NN) as non-linear discriminant functions for segmenting between driveable and non-driveable areas in a scene. The main reason is that in unstructured environments it is often hard to develop well working methods in a traditional formal way since the knowledge about the characteristics of the terrain is limited. The NN’s ability to learn from example is then of great interest. Furthermore NNs often possess an ability to generalize well and thus result in a robust system.

2.6.1 General

Neural networks have been used in a wide variety of applications. The advantages that are of particular interest for our purposes are the ability to learn from training, good generalization capability and high processing
speed when implemented in appropriate hardware. The disadvantages are mainly that the training time rapidly increases with the size of the NN and that the training material must be chosen with great care. Additionally it is hard to understand the function of a particular NN and to debug it if it does not perform as desired.

Two completely different approaches can be taken in learning, supervised or unsupervised learning. In supervised learning the learning algorithm is constantly presented the correct output for comparison, while in unsupervised learning the learning algorithm has to discover relevant features in the training set on its own. In this application, there is a correct output available and supervised learning is therefore chosen. For road segmentation, the most suitable supervised learning principle is probably error back-propagation. It is straight-forward and well suited for this application, since we can provide a correct output for every input.

Of course a neural network can not perform better than the information provided to it permits it to. The choice of information is here very important. There are two main schools of thought for what to present to the network. One believes that only the most important features of the material, obtained by extensive preprocessing, should be used, while the other claims that feature extraction should be left to the neural network, i.e. the network should be fed raw data [Pomerleau 1993]. We will use a combination of the two, some preprocessing representation will be done by extracting a suitable colour representation, but the rest of the work will be left to the NN.

2.6.2 Architecture

A simple feed-forward neural network (FFNN), trained using error back-propagation, is the most straight-forward approach in neural network computing and if it works, it is probably the best solution. For more complex NNs might provide better results.

2.6.3 Training

The choice of training algorithm naturally depends on the chosen architecture. For the FFNN, error back-propagation will be used.
However, there are many different versions of back-propagation which all have their advantages and drawbacks.

The demands of training increase rapidly with the complexity of the neural network, and it is therefore necessary to keep the network as simple as possible. Information reduction before the neural network retina stage, a hidden layer of limited size and a non-complex output are therefore desired.
3

Proposed system structure

We intend to implement a method the show the usefulness of colour for scene interpretation. The input consists of 2D RGB image sequences from a standard consumer CCD camera that has been digitized and stored in a computer. These sequences will be processed by our system and the output will be presented as a sequence of maps of the terrain in front of the vehicle.

Figure 3.1 shows a typical scene from the operation scenario along with a suitable output. In addition, we would, however, like some sort of confidence measure for each area. How certain are we that an actual area actually is what we think it is?

![Image](image.png)

Figure 3.1. A typical scene with a suitable output map.

Treating the image by blocks would result in two advantages compared to treatment pixel by pixel – the workload will be less and we will obtain increased robustness to variations in individual pixel values. An obvious disadvantage is decreased map resolution. The block size can be adjusted to arrive at a suitable compromise.
Proposed system structure

For the segmentation we follow the standard procedure discussed in 2.2 Segmentation, with feature extraction, discrimination and the actual segmentation. The feature extraction will be a critical part. If we succeed in finding a suitable feature, the rest of the segmentation will be much easier and the system performance better.

We propose a general system structure of 8 parts according to Figure 3.2. The exact content of the different parts is to be decided later after having explored possible solutions (see 4. Exploration of possible solutions).

Figure 3.2. Overall block diagram of the system.

- Preprocessing can incorporate miscellaneous types of preprocessing of the images. Information reduction through down-sampling is probably necessary.
- Feature extraction. Colour and texture will be considered.
- Discrimination of the extracted features will be done block-wise to save time as well as to obtain increased robustness. Considering entire blocks minimize uncertainty due to variations in individual pixels.
- Segmentation of the driveable area to minimize the information to pass on to the path planning algorithm. Only accessible areas are of interest.
• A confidence measure is very important for safe driving - the vehicle needs to know how much it can trust its inputs. The confidence will be based on information about the current frame as well as on correlation between blocks in consecutive frames (stored in a buffer). Calculations will only be done for segmented driveable areas to save time.

• The output, the driveable area and the confidence measure, will be mapped onto a model of the world in front of the vehicle. This part will not yet be implemented since no 3D information of the scene is available. Mapping onto a flat world model would be trivial and would not add any information.
4

Exploration of potential solutions

4.1 Operating scenarios

The operating scenarios consist of 4 test runs with a total time of 25 minutes, where the vehicle is driven by remote control off-road in a tropical jungle terrain.

![Block diagram showing the capturing of the operating scenarios.](image)

The image sequences were collected using a standard consumer Sony video camera, mounted on the vehicle, and then captured using a Matrox Media XL video capture card in 320x232 resolution according to Figure 4.1. Due to technical problems, we were not able to retrieve continuous image sequences, so we will only have a test set of still images from the operating scenarios.

The existing scenarios were judged sufficient for a proof of principle. The terrain is diverse and includes narrow road segments as well as open areas, both lined with grass, bushes, trees and a lake. Shading from trees is also present. Please refer to Appendix A for examples of the scenes. It
would, however, still have been interesting to evaluate the proposed method on an even more diverse material, that could include different lighting and weather conditions and more complex shading.

### 4.2 Experimental method

The potential solutions will be tested on the operating scenarios and their performance will be evaluated.

All algorithms are implemented in MatLab v.5 code (m-files). These are structured according to the block diagram in Figure 3.2 and each block in the diagram is implemented as a single m-file. For description and a block diagram of the individual files, please refer to the section describing the corresponding function.

For exploring the different ideas we created a test set of 3850 blocks of 16x16 pixels of known nature. The set was generated by selecting 385 blocks of 32x32 pixels and then from each extract 10 smaller blocks by random selection and mirroring. The set is meant to properly reflect the operating scenarios but it only contains “clear-cut cases”, i.e. typical cases that are either driveable or not. No “border cases” were added. The training set contains mostly relatively “easy cases”, since a large majority of the blocks in a scene are fairly easy to classify, only a minority cause problems. The original 385 blocks can be seen in Appendix B.

Initially, we explore the different image features and conclude which one is most suitable. This feature is later used for the exploration of the discrimination methods.

### 4.3 Relevant image features

#### 4.3.1 Colour

As expected colour proved to be a good feature for discriminating between red soil and green vegetation. However problems naturally arise when objects in the scene deviate too much from these ideal colours. Many objects that are redish, are not road, for example dry leaves and some man-made objects. Conversely, slightly green areas in front of the vehicle that could very well be driveable will be classified as non-
driveable. Without additional information, it is clear that two objects with the same colour will be classified as having the same properties.

Invariance to lighting conditions is an important property and we therefore seek representations that are as invariant to the lighting intensity as possible. However shading also results in a colour shift (refer to section 5.3.2 Invariance to lighting conditions). The “direction” of the shift depends on the scene and the colour of the surfaces lighting up the shaded area. The shaded area will no longer be lit by the white sun light but by reflection from other objects and hence the perceived colour of the shaded area will be slightly shifted towards the colour of the light that these objects emit, i.e. the colour of these objects. In ALVINN, the only object illuminating a shaded part of a road is usually the blue sky, hence the reported colour shift towards blue. In our application shading is mostly due to trees and the road segments are then lined with green vegetation. The sky is however still there, so the colour shift of the shaded area will be towards a mix of green and blue. Figure 4.2 shows the colour shift for typical shading of a driveable area in a scene. The non-shaded area is more red than the shaded one, whose colour is shifted towards a mix of green and blue.

Figure 4.2. Scatter diagram for pixels from shaded (blue) and non-shaded (red) driveable region. The brightness component has been removed and this only shows the “colour shift”.

To truly compensate for this shift, we would probably need much a priori knowledge about the scene, which is unavailable, and we therefore focus on finding a representation that minimizes its influence.

**Normalized RGB**

Normalized RGB is frequently used to decrease the influence of lighting. When just normalizing the three RGB components and displaying them as an image, shadows are still very visible, which indicate variance to lighting conditions. Like in the ALVINN project [Pomerleau 1993], the components can be used individually in ad hoc solutions for segmenting the scene. There, they showed that the blue component can be made fairly robust to shading, but blue will not be suitable for our application, since we are to classify between mainly red and green areas.

![Normalized red](image1.png) ![Normalized green](image2.png) ![Normalized blue](image3.png)

**Figure 4.3.** Histograms of normalized RGB values for driveable areas (red) and non-driveable areas (green).

Figure 4.3 shows histograms for normalized RGB values to illustrate there ability to discriminate between driveable and non-driveable areas. The histograms are based on 2 x 179200 pixels from the test set. All three components show unseparable groups, but normalized green seems to be the best criterion for the classification.
Exploration of potential solutions

Figure 4.4. Histogram of normalized green values of a shaded (blue) and a sun lit (red) driveable area.

Figure 4.4 shows the same type of histogram for normalized green and a shaded area (blue) and a sun lit area (red). The colour shift is clear even when using normalized green, so the conclusion is that normalization of RGB does not eliminate the effects of shading.

Another reason for not basing our method on normalized green is that it is a very ad hoc solution, that happen to work reasonably in this particular case, but with a somewhat different terrain we might need to completely change our approach.

Hue

The block-wise extraction of hue is implemented in an m-file according to the block diagram in Figure 4.5.

![Block diagram of the transformation from RGB to hue](image)

Figure 4.5. Block diagram of the transformation from RGB to hue (implemented as an m-file).
Exploration of potential solutions

Hue is best represented as a circle and a discontinuity is hence unavoidable when using a one-dimensional angular representation. This problem can somewhat be avoided by placing the discontinuity in a region where few or no pixel values occur. This is done by applying a rotation to the RGB-to-YBPbPr transformation matrix. Blue-magenta was chosen for the current operating scenarios. Figure 4.6 illustrates the chosen hue representation with hue values in the range [-1,+1]. Other possible representations would have been complex numbers or modulo-$2\pi$-arithmetic. The latter would, however, have resulted in complications in the block averaging process.

![Figure 4.6. Hue representation](image)

![Figure 4.7. A driveable area with clear gray parts.](image)

Hue is suitable for representing saturated colours, but is undefined for grayscale values. This is unfortunate, since gray very well may occur in a natural scene, for instance as gravel and small stones in the road as can be seen in Figure 4.7, and therefore we would like a meaningful representation. There is however no simple solution, but this will have to be solved ad hoc depending on the application. One may argue that completely gray pixels are rare, since we are dealing with natural scenes there is often some small amount of colour present. This is true, but even for these pixels it would be very unfortunate to use hue. Noise occurs in the image and without studying its nature further, let us assume that it is simple additive noise. For lowly saturated pixels the signal-to-noise ratio will then be very low and the hue output will be more or less random. A most unreliable basis for successful classification! The same reasoning can be done for pixels with too low intensity. Hence both
Exploration of potential solutions

pixels with low saturation and low brightness, below some threshold value, should be considered as indeterminate.

![Image of color vision and hue for autonomous vehicle guidance](image)

**Figure 4.8.** The hue output for different choices of threshold. White corresponds to indeterminate pixels.

The threshold value is set empirically and is a trade-off between noise reduction and eliminating useful colour information. Since it is hard to optimize this trade-off in a formal way, it is performed by visually determining a proper setting, by studying the amount of visible noise in hue and the amount of pixels classified as indeterminate in an area with low saturation and intensity. See Figure 4.8 for an illustration. The upper image shows the context of the block that is converted to hue representation. The left image contains too much noise, which can be seen as magenta and yellow pixels, due to a too low threshold. The middle one corresponds to a too high threshold, since much important
colour information is lost by labeling classifiable pixels as indeterminate. The right image shows the result when using the chosen threshold.

From this it follows that methods based on hue will perform better in colourful surroundings and less well on, for instance, paved roads and in between concrete buildings.

Figure 4.9. Histogram of hue values for the same set (driveable - red, non-driveable - green) as was used with normalized RGB in figure 4.3, with the addition of a subset of blocks of sky (blue).

As expected hue proved to be a good criterion for classifying between driveable and non-driveable areas in the operating scenario, as can be seen in Figure 4.9. Compared to normalized RGB, hue discriminates better between the different groups and is hence a better representation. The groups are, however, not separable, due to the fact that there will always be some rare dispersed green pixels in a road and conversely some red pixels in the non-driveable area. However, we can hopefully minimize their influence by treating entire blocks at a time.

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Driveable</th>
<th>Non-driveable</th>
<th>Sky</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>-0.3977±0.0003</td>
<td>-0.0814±0.0005</td>
<td>0.7470±0.0010</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>0.0589±0.0002</td>
<td>0.1041±0.0004</td>
<td>0.0658±0.0007</td>
</tr>
</tbody>
</table>

If we assume gaussian distribution of the groups in Figure 4.9, we can use statistical methods. The mean and standard deviations are presented
in Table 4.1 with their corresponding 95% confidence intervals. Pixels with low saturation were not included. The confidence intervals are narrow due to the large sets (more than 150 000 pixels per group for driveable and non-driveable). When seen as gaussian distributions, the groups are well separated.

Figure 4.10. Histogram of hue values of a shaded (blue) and a sun lit (red) driveable area.

When extracting only the hue information and discarding intensity and saturation, we obtain a method which is robust to shading effects. However colour shift cannot fully be compensated for without more complex approaches, but fortunately the effects of the colour shift is limited when using the hue representation as can be seen in Figure 4.10 (see also Figure 4.2). The hue average for the block barely changes, even though the histograms are not identical. The difference in shape between the two histograms is probably not due to the presence of absence of shading, but is due to that the fact that the underlying structures of the blocks are not identical. This robustness to lighting conditions is a very important finding and a much desired property for our application.

Comparison between normalized RGB and hue

From the results above it appears that the hue representation is superior to the normalized RGB for our application. Not only does it separate the groups better, but it is also more robust to shadows. But the experiments
were done on test sets of individual blocks and not on a real scene. Let us do a comparison for a real scene.

![Image](image)

Figure 4.11. Suppression of shadow using different representations.
In the scene shown in Figure 4.11, it is very hard to segment the road from the rest due to the shadow across the road. As can be seen in the upper right image, most approaches based on grayscale images will probably fail, especially if they involve plain thresholding.

Then it is better to use colour, since there is a clear colour difference between the road and the rest. Our use of colour appears to be a correct choice, but the shadow is very distinct in the colour RGB image. We want a representation that is invariant to lighting conditions, such as shading. Normalized RGB is one attempt to obtain this, but the shadow is still clearly visible as can be seen in the lower left image in Figure 4.11. In the lower right image the image is represented using hue and the shadow is still a bit visible but to a much lesser extent than for the other representations. Hue is nearly invariant to shading and once again hue appears to be the superior representation.

This result is very encouraging since invariance to changing lighting condition is a major issue in computer vision. It would be very interesting to test the hue representation for various other types of working environments to see if this could be a general method for obtaining invariance to lighting conditions. Even if it would fail to be a universal method, it is however likely that we can find many other interesting application areas.

4.3.2 Texture

It is of course a limitation to only consider the colour information of the scene and in an attempt to improve the performance of the method we wanted to incorporate some other relevant feature. Texture is a very important feature for human vision and therefore seemed like a natural choice for improving the robustness of the method. Combining colour and texture should form a very solid basis for correct scene interpretation. Our method based on colour has proved good to segment between red mud and vegetation, but fails when colour is not a relevant feature for the segmentation. Especially for distinguishing between red soil and shallow red water, texture should provide improved discriminating power.

Natural scenes are very diverse and it would therefore be very time consuming, if not impossible, to implement a matching technique to cover all possible cases that might arise. This would not go well with the
demand for a real time application. Instead, I chose to study the frequency spectrum of the intensity within blocks of the image. A smooth surface, such as water, should ideally correspond to a dirac distribution while textured areas should present a more complex spectrum in the frequency domain. With this reasoning, the variance of the spectrum would provide some sort of measure of the amount of texture found in the block. Different variations around this basic idea have been tested, but several problems arose and the outcomes were poor.

![Histograms of within-block-variance for the frequency spectrum of grayscale blocks, of ground (blue) and water (red).](image)

Figure 4.12 shows histograms for two slightly different measures of within-block-variance for the spectrum of grayscale blocks. The left is the variance of the complete spectrum, but here we risk that high frequent noise has a big influence on the measure. The blocks of water have even higher variance than the blocks of ground, which was not what we expected. An attempt to remove the noise is to disregard high frequency components and this is done in the right histogram. The result is better with water having lower variance than ground, but still there is a big overlap and it is impossible to separate the groups. Furthermore, when comparing the frequency spectrums for water and ground they are often surprisingly similar as can be seem in the two examples in Figure 4.13. The left plot shows the 2D frequency spectrum for water and the right plot is for ground. The left corner of each plot corresponds to low frequency components corner (the zero component has been removed), while high frequency components are found in the right corner.
Exploration of potential solutions

Figure 4.13. Frequency spectrum for two blocks of water and ground, respectively.

We have identified the following reasons for the unsatisfactory results:

- Changing nature of the surfaces – Ground as well as water both show very diverse textures and can both appear as smooth or rough and are hence hard to separate. Figure 4.14 and Figure 4.15 show how similar two blocks of different types can appear even to a human observer.

- Different scale - The scale depends of how far away the surface is and hence the perceived spectrum varies.

- Low resolution - For distant surfaces, the resolution will prove insufficient to pick up textures and the method will fail. A way to avoid this problem would be to only implement the method on areas within say 5 metres ahead of the vehicle.

- Focus – Without sharp images one cannot expect to reliably detect textures in the scene. Unreliable focusing of the camera used or vehicle vibration, due to rough terrain and a slow capture rate in the camera, may both result in blurry images, where texture information is partially lost, or completely.

- Slow – Even though there are efficient hardware implementations of FFT commonly available, this method will probably decrease the possible frame rate. It can be run parallel to the colour discrimination module, but it risks slowing down the processing.

Some of these might be possible to overcome by further studies, but the outcomes are uncertain. Considering the quality of the presently available
operating scenarios, especially the unreliable focus, this prevented us from obtaining a robust method and we abandon further exploration of texture.

4.3.3 What more?

Humans are obviously capable of classifying between mud and vegetation even under difficult conditions. So what does a human possess, other than an ability to interpret colour and texture, to distinguish between the road and the rest? Well, above all we have an understanding of what is currently in front of us, we know how these kinds of terrain are supposed to look like, i.e. we have knowledge about the context. Figure 6.8 and 6.9 illustrate the importance of context knowledge. In Figure 6.8 it is hard to understand what type of terrain the blocks contain, but with the context from Figure 6.9 we easily interpret them.

Figure 4.14. Two small blocks with no context. Hard to interpret.

Figure 4.15. The context for the blocks in Figure 4.14. With their context the blocks are easily interpreted.
Without giving this ability to the machines, they will probably always lack in robustness compared to a human observer.

However we believe that hue will provide us with enough information for obtaining a well functioning system and from here on we base our method on hue.

### 4.4 Discrimination method

After having extracted the interesting features, these need to be processed through discrimination to find the feature characteristics and map this onto a representation that can be used easily to determine group membership. The discrimination method will be explored based on the hue representation of the images.

#### 4.4.1 Neural network

![Diagram of discrimination using a neural network](image)

Only feed forward networks (FFNN) were taken into consideration for mainly two reasons - they are good for quick processing and the classification task is considered fairly simple. Different architectures with varying number of layers and hidden units were tested.

Using both hue and FFNN results in a problem. Since there is no hue representation for gray, any gray pixels need to be treated separately, but FFNN use a retina of fixed size, i.e. they need a fixed number of inputs. A possible, but perhaps not satisfactory, solution is the assignment of a “dummy hue” to these gray pixels, in this case blue-magenta was chosen since it does not occur naturally in the operating scenario. For safety reasons the hue was set to a value of +1 to ensure that the pixel is classified as non-driveable.
Exploration of potential solutions

We used MatLab’s Neural Network Toolbox [Neural Network Toolbox 1998] to implement a neural network method according to the block diagram in Figure 4.16. We chose a sigmoid transfer function, a common choice in multi-layer networks, and elaborated with different structures and training algorithms. The resilient back-propagation learning algorithm is especially well suited for nets that use sigmoid transfer functions and it showed good performance compared to more traditional back-propagation algorithms.

The neural network training was done on two thirds of the training set of 3850 blocks while the last third was left for evaluation. The 2566 blocks (two thirds of 3850 blocks) were selected at random.

Table 6.1 shows the results for some different structures and different number of training runs. Each run involves training on all of the 2566 blocks. Each network has an input layer of 16x16 input units, one for each pixel value, and a one element output layer. The “structure” column describes the number of elements in the layers after the input layer. 7-1 means one hidden layer with 7 units and one output unit.

Table 4.2. Result for the training of neural networks for some different structures and training iterations.

<table>
<thead>
<tr>
<th>Network structure</th>
<th>Number of iterations (x2566 blocks)</th>
<th>Mean square error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>20</td>
<td>0.0968</td>
</tr>
<tr>
<td>0-1</td>
<td>50</td>
<td>0.0784</td>
</tr>
<tr>
<td>3-1</td>
<td>20</td>
<td>0.0663</td>
</tr>
<tr>
<td>3-1</td>
<td>50</td>
<td>0.0623</td>
</tr>
<tr>
<td>5-1</td>
<td>20</td>
<td>0.0683</td>
</tr>
<tr>
<td>5-1</td>
<td>50</td>
<td>0.0570</td>
</tr>
<tr>
<td>7-1</td>
<td>20</td>
<td>0.0654</td>
</tr>
<tr>
<td>7-1</td>
<td>50</td>
<td>0.0655</td>
</tr>
<tr>
<td>7-1</td>
<td>100</td>
<td>0.0530</td>
</tr>
<tr>
<td>20-1</td>
<td>100</td>
<td>0.0547</td>
</tr>
</tbody>
</table>
Exploration of potential solutions

We hoped that the NNs would identify other interesting features than just the hue average in the blocks. For fairly simple NN structures, like ours, we can study the NN weights from the input retina to each of the hidden units to identify which kind of textures that particular hidden unit is sensitive to [Pomerleau 1993]. Figure 4.17 shows the weights to the three hidden units of a 3-1 FFNN. The weights range approximately [-0.3, +0.5] and are represented by grayscale values in the images. There is no clear structure in the weights since the pattern seems more of less stochastic. This indicates that the NN has found no particular texture in the training set that is relevant for the classification task. Note that this is based on the hue representation and not the grayscale value as in section 4.3.2 Texture.

![Figure 4.17. NN weights from input layer to hidden units.](image)

The results from Table 4.2 seem very promising, but if we are faced with a scene with pixels of low saturation, such as small gray stones lying on top of the red soil as in Figure 4.7, the network runs into problems (see Table 4.3). Figure 4.18 shows four RGB blocks (left) with their corresponding hue representations (right) taken from a similar scene. The area is mainly red and covered by gray stones of size 5-10 cm and is considered driveable. Gray pixels (too low saturation) have been assigned a “dummy hue” (value +1), here represented by white. However, only one of four blocks are classified as driveable and this is of course unacceptable.
Exploration of potential solutions

Figure 4.18. Blocks from scene containing gray stone of size 5-10 cm. The area is mainly red and considered driveable. RGB blocks with their corresponding hue representations. Gray (too lowly saturated) pixels have been assigned a “dummy hue”, here represented by white.

Table 4.3. Discrimination of the blocks in Figure 6.4.1.2 using a neural network. The desired discrimination output is +1. Only one of four blocks are correctly classified.

<table>
<thead>
<tr>
<th>Discrimination of block</th>
<th>Neural network output (+1=road, -1=nonroad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper left</td>
<td>-1.000</td>
</tr>
<tr>
<td>Upper right</td>
<td>-1.000</td>
</tr>
<tr>
<td>Lower left</td>
<td>0.997</td>
</tr>
<tr>
<td>Lower right</td>
<td>-1.000</td>
</tr>
</tbody>
</table>

Note: The NN is 3-1 and normally performs well with a MSE of 0.059 on our standard test set.

Table 4.2 shows that neural networks perform well in discriminating between driveable and non-driveable regions for well-saturated scenes, but Table 4.3 shows that the system is not robust to disturbances in the form of lowly saturated regions as in Figure 4.18.
4.4.2 Block averaging with smooth thresholding

![Diagram of block averaging and smooth thresholding](image)

Figure 4.19. Block diagram of the discrimination using block averaging and smooth thresholding (implemented as an m-file).

Block averaging can be seen as a special case of the FFNN and should hence be more limited in its performance, but it possesses one additional and very valuable property - it can be used with a variable number of inputs. We can disregard lowly saturated pixels and only consider the coloured ones.

This proves to be useful since lowly saturated pixels frequently occur in both driveable and non-driveable areas. By using this method we disregard these gray unclassifiable pixels and look at the surrounding coloured ones in the block, an intuitively very appealing solution to avoid this problem associated with the hue representation.

When seeing block averaging as a special case of the FFNN, it is natural to also adopt its smooth threshold function. The smooth threshold function is intuitively appealing compared to a discrete one, since it directly, in its output, provides an indication of confidence. This will later be used for deriving a confidence measure (see section 6.6 Confidence measure).

We choose a sigmoid function as our smooth thresholding function. There are two parameters for the function that need to be set. These are the threshold and smoothness parameters, that were set empirically to -0.26 and 0.0035, respectively, by minimizing the mean square error for classification of the training set. They can also be set based on the group statistics of the test set, presented in Table 4.1. If we assume equal probabilities for the driveable and non-driveable groups and leave out the sky group, Bayes’ classification gives a threshold of -0.2726 based on the statistics derived in Table 4.1. A suitable smoothness parameter can be found by studying the probabilities of correct classification as discussed in section 2.4 Smooth thresholding. However, we prefer...
Exploration of potential solutions

keeping the empirically derived parameters due to the fact that we trust the mean square error more than the assumption of a gaussian distribution.

The block averaging with a smooth threshold function will be implemented in an m-file according to the block diagram in Figure 4.19.

The block averaging performs less well than the neural network on the standard test set with a mean square error of 0.0851 compared to 0.0530 for the neural network. The difference is, however, not considered vital since both methods actually perform very well. There is, however, a significant difference when treating the lowly saturated test set shown in Figure 4.18. As can be seen in Table 4.4, the block averaging method still performs very well, while the neural network approach failed (Table 4.3). All four blocks are here correctly classified.

Table 4.4. Discrimination of the blocks in Figure 4.18 using a block averaging and smooth thresholding method. The desired discrimination output is +1. All four blocks are correctly classified.

<table>
<thead>
<tr>
<th>Discrimination of block</th>
<th>Discrimination output (+1=road, -1=nonroad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper left</td>
<td>0.9987</td>
</tr>
<tr>
<td>Upper right</td>
<td>0.9985</td>
</tr>
<tr>
<td>Lower left</td>
<td>0.9997</td>
</tr>
<tr>
<td>Lower right</td>
<td>0.9991</td>
</tr>
</tbody>
</table>

On the training set, the block averaging performs slightly worse than the FFNN, but this was expected since it is a special case. However, Table 4.4 shows that using block averaging results in a method that is much more robust to lowly saturated regions, which is a valuable property. In the light of this robustness, we believe that not only is block averaging an easier method, but it is also superior in performance to the neural network, for this application.
4.5 Segmentation

The segmentation of the discrimination output is more or less trivial, since the complicated part of the thresholding is done in the design of the discrimination function. The segmentation is done by plain thresholding and grouping of the driveable areas.

The discrimination method is designed to output a value of +1 for driveable areas and -1 for non-driveable ones and zero signifies equal probabilities. Hence the threshold is set to zero, above is driveable and below is non-driveable. The result is a binary image.

Only driveable areas accessible to the vehicle will be considered. We define “accessible area” as a driveable area being connected to the bottom line of the image, the area immediately in front of the vehicle. The image is then segmented using row-wise convolution with two standard segmentation kernels, which is a standard procedure for segmenting binary images.

4.6 Confidence measure

Even though we trust our system in most cases, there will be occasions when we risk poor or even erroneous output from the system and we need to detect these cases. Ideally we would probably want the Bayesian probability for a correct output. In practice, this is clearly impossible and we have to settle for more approximate estimations of confidence.

The confidence was found to depend on the following quantities:

- The output from the smooth classification function. Values near the threshold correspond to uncertainty on how to classify. This is based on the assumption that the classification function correctly captures all dimensions of the problem. However it can be dangerous to let a method assess itself.

- A special within-block variation by computing var\{X>threshold\}. If the block is uniform and all pixels are on the same side of the threshold, it is likely that the classification will be simple and correct.
• Saturation. High saturation corresponds to “a strong and clear signal” and the input is likely to be correct and not very distorted due to noise or natural lighting effects.

• The nature of neighboring blocks. A block is more likely to be of the same type as the ones surrounding it.

It is unfortunately not possible to obtain an objective measure of how the method actually performs, due to the nature of the problem, so these measures are all a matter of probabilities. They focus on detecting cases where we know the method performs less well, some are due to the choice of method and some to the nature of the problem.

The last component, the nature of neighboring blocks, is somewhat different from the other three and deserves special attention. Only still images are available for testing. The correlation between frames decreases as the time between them increases and since the still images have 3 seconds or more between consecutive frames, we can not expect much correlation. Therefore we will not consider the relationship between blocks in consecutive frames, only between blocks within one frame. Without statistically evaluating the relationship between neighboring blocks, we conclude that it is more likely that a block is of the same type as its neighbors than the opposite. We assign an ad hoc formula that at least approximately captures this relationship, according to

$$\text{conf}^4 = \frac{1}{8 \text{ neighboring blocks}} \sum_{i} s_i \quad \text{where} \quad s_i = \begin{cases} 1 & \text{if block is of same type} \\ 0 & \text{if block is of opposite type} \end{cases}$$

There are a total of 8 neighboring blocks and the measure must therefore be normalized by division by 8.
Figure 4.20 is a block diagram of the proposed method for obtaining a confidence measure. Initially a single confidence measure for each one of the criteria is calculated and these are then weighted together. Missing out on any one of them is enough to decrease the confidence of the classification, i.e. all have to be fulfilled to have high confidence. Hence some sort of analogue AND, i.e. multiplication, is suitable. To achieve proper weighting between the four, each one is normalized to [0,1] prior to the multiplication.

\[
\text{confidence} = \left( \text{conf}_1^a \times \text{conf}_2^b \times \text{conf}_3^c \times \text{conf}_4^d \right)^{(a+b+c+d)}
\]

The formula is written in a general form where the influence of the different individual confidence measures can be adjusted by changing the a, b, c and d powers. This approach shows encouraging results. However, we have found that equal weighting of the four results in good performance, but we still keep the more general form to facilitate future adjustments. Please refer to section 6.1. Performance of the system for examples of using the confidence measure.
Proposed method

Based on the results from section 4. Exploration of potential solutions, we propose a system structure according to the block diagram in Figure 5.1. We will use the hue information of the images for the scene interpretation, since we have shown that it is the most relevant and suitable representation. Block averaging with smooth thresholding will be used, since it is a simple solution, it can disregard gray pixels and its performance is comparable to that of the FFNN. The suggested confidence measure will also be used since it appears to successfully capture indications of low confidence. The buffer will not be used at this moment, since we have not been able to make use of continuous image sequences and the correlation between consecutive frames. This should however be implemented in the future and we therefore keep the buffer in our block diagram.
6 Evaluation

6.1 Performance of the system

Due to the unstructured nature of the problem, it is hard or even impossible, to objectively evaluate the performance of the method. To say where the driveable area starts and ends is a matter of definitions, since there is often a smooth change from driveable to non-driveable, and will most probably vary depending on the observer. It is hence more or less impossible to present an absolute quantitative measure of performance.

From the testing of the discrimination method above (see 4.4.2 Block averaging with smooth thresholding) we have proof that the method performs well in “clear cases”, i.e. for typically driveable and non-driveable blocks. The mean square error is 0.0851. For more unstructured scenes, where blocks are more difficult to classify even for a human observer, we do not have a correct answer and the evaluation will hence be subjective.

We believe the best way to test the system’s performance is to run it on real images from the operating scenario and study the output subjectively.
Evaluation

Colour vision and hue for autonomous vehicle guidance
Figure 6.1. (opposite side) Four examples of segmentation.

Figure 6.1 shows four examples. The top images of each triplet are the original 320x132 pixel RGB images. The middle ones are the hue representation, the saturation and intensity have been taken away (normalized). The bottom images are the segmented output, where the intensity represents confidence.

The result appears to be good, the two left scenes are correctly interpreted while the method as expected runs into some difficulties for the right ones. The upper right scene is difficult to interpret, even for a human observer. “Is it green or is it not?” The hue representation reveals that there is some of both red and green and that is why the output is somewhat uncertain and not as clear as for the left scene. The water appears more red than the rest, which is perfectly normal, since this is water mixed with red mud. Identification of this kind of puddles would be of interest but is not possible with only hue. The lower right scene is also hard to interpret based only on colour, since the shallow water appears to be red due to the fact that we see the red soil underneath. Only based on colour, we can not expect a reliable output for these types of cases, we need complementary information from other sensors. Note, however, how the shadow in the left image practically disappears in the hue representation. Hue is indeed robust to changing lighting conditions. Further examples of system performance can be found in Appendix C.

It is hard to tell if the confidence measure is optimized or correct, but by studying the above results we can see that it at least gives an indication of whether the output is reliable or not. The left scene is interpreted with great confidence while the confidence is lower for the right scene. Without a thorough statistical investigation of the image material we will have to settle for this subjective evaluation.
6.2 Fulfillment of basic criteria

6.2.1 Robustness

The method appears to be robust to different disturbances with one major exception, colour and hue. For natural reasons the system is sensitive to changing hue, and noise or other disturbances in hue can therefore cause problems. Typically structured noise like red objects outside the driveable areas and green ones within them frequently cause problems. Examples of problems are:

- Man-made objects
  - Red dust in the air, on objects or even on the camera lens.
  - Red muddy water or shallow water where the red soil beneath is visible.
  - Dry leaves and vegetation.

This once more highlights the demand for complementary information from other sensors.

However the method is robust to shading compared to previously seen systems and this is a very important property. It would be interesting to test the performance under more complicated shading conditions on an extended operating scenario.

6.2.2 Speed

The algorithms have only been implemented as MatLab scripts and an absolute measure of execution speed for real-world hardware implementation is therefore not available. However estimations can be made with knowledge of the complexity of the method and available hardware.

For 320x128 pixel images, the implementation requires approximately 300-800 kflops/frame, depending on the size of the driveable area to segment and how much down-sampling that is done before the segmentation.

This is believed to be satisfactory and the method can be implemented in a real-time application.
6.3 Further evaluation

The evaluation has been carried out on a limited set of operating scenarios. A more diverse set would be of great value for assessing the usefulness of the proposed method. For instance, testing in a slightly different terrain and under different lighting and weather conditions would be interesting.
7

Discussion

7.1 Applicability

This is the first step in deriving a usable method for visual guidance of autonomous off-road vehicles in the described operating scenarios. It is to be seen as a proof of principle and the method may need adjustments for efficient hardware implementation for real-world testing. However, the principles have been shown successful and should therefore hold.

The method has been developed for and evaluated on a relatively limited operating scenario, but it is most likely possible to apply it in other types of terrain with similar conditions, with minor adjustments. We believe that the method is flexible and parameters can easily be adjusted to adapt to changing conditions. As discussed above, a main demand is that colour has to be present and a relevant feature for the scene interpretation. Hence predominantly gray terrain like paved roads and concrete buildings can not be dealt with.

Further improvements as well as integration of other sensory systems is probably needed for a truly functional system. Considering only 2D colour information is a severe limitation that restricts the performance and other complementary sensors are needed for reliable scene interpretation.

7.2 Future improvements

I believe improvements can be done in the domain of reliability and robustness by studying dependence and correlation between adjacent areas in both time and space.

However the most interesting future improvements go beyond simple 2D classification of single frames to derive additional information. The most desirable addition to the current solution would be to add 3D
information in some form, either by using a laser range finder, stereo cameras or movement in a single image sequence.

Furthermore, advantages can be made by successfully merging the system into a more complex multi-sensory one.
Conclusions

The aim of this project has been to study the usefulness of colour for identifying driveable areas in a specific operating scenario. The usefulness has been shown by the successful implementation of a method for the task.

We have showed that hue is both robust to changing lighting conditions and a relevant feature for interpreting scenes from this kind of terrain. In addition to hue extraction, the system will, among other things, include classification, by block averaging and a smooth threshold function, and estimation of confidence. The suggested method is both simple and flexible, parameters can easily be adjusted to adapt the method to different operating conditions.

However colour and hue are not enough for correct and reliable scene interpretation, complimentary systems based on other sensors are needed. It is for instance a limitation to not have 3D information, but we believe the method easily can be extended to incorporate this. We also believe that in the future the here proposed system can easily be combined with others to form a reliable system for visual guidance in unstructured off-road terrain.
References

http://bruce.cs.cf.ac.uk/bruce/Colour_recognition/colour_recognition_text.html


References

http://www.cs.cmu.edu/afs/cs/project/alv/member/www/projects/ALVINN.html


[UniBwM] University of the Federal Armed Forces Munich (Universität der Bundeswehr München, UniBwM). http://www.unibw-muenchen.de/campus/LRT/LRT13/index_E.html


Appendices

Appendix A – Examples from the operating scenario
Appendices

Appendix B – The training set
Consisting of 385 blocks of 32x32 pixels.

Driveable areas:
Non-driveable areas:
Appendix C – Examples of segmentation

The left column shows the original scene to be segmented and the right column illustrates the corresponding segmentation output. As before, the intensity in the segmented image represents the confidence measure, the brighter the more confident. The input images are 320x128 pixels RGB.
Appendices

Colour vision and hue for autonomous vehicle guidance