# IDENTIFICATION & SEGMENTATION OF LAWN GRASS BASED ON COLOR & VISUAL TEXTURE CLASSIFIERS

by

ALEXANDER SCHEPELMANN

Submitted in partial fulfillment of the requirements

for the degree of Master of Science

Thesis Advisor: Dr. Roger D. Quinn

Department of Mechanical Engineering

CASE WESTERN RESERVE UNIVERSITY

August, 2010

**CASE WESTERN RESERVE UNIVERSITY**

**SCHOOL OF GRADUATE STUDIES**

We hereby approve the thesis/dissertation of

Alexander Schepelmann

candidate for the Master of Science degree *.

(signed) R. D. Quinn
(chair of the committee)

Francis Merat

Michael S. Branicky

(date) 05/24/2010

*We also certify that written approval has been obtained for any proprietary material contained therein.

# Table of Contents

## List of Tables

## List of Figures

## Acknowledgements

I would like to thank:

Rick Hudson and Frank Merat for teaching me about the latest computer vision and data analysis techniques and for the innumerable meetings we have had over the years, without which this thesis would not exist.

My advisor, Roger Quinn, for allowing me to work on this project and advising me while I was at CWRU.

Michael Branicky, who took me on as a TA for ENGR 131 during my sophomore year and who introduced me to Dr. Quinn and the CWRU Cutter project.

MTD Products, Inc. for sponsoring this research.

The CWRU Cutter team for all of our ridiculous adventures.

All of the members of the Biorobots lab for their help, advice, and humor.

Case Western Reserve University for allowing me to get involved in research, TA, and teach.

Above all, my parents, Heinz and Carla, and my fiancée, Grace, for their never-ending love and support, and for encouraging me to follow my dreams.

This work was partially funded by the Case Western Reserve University office for the Support of Undergraduate Research and Creative Endeavors (SOURCE).

# Identification & Segmentation of Lawn Grass
# Based on Color & Visual Texture Classifiers

## Abstract

By

ALEXANDER SCHEPELMANN

CWRU Cutter is an autonomous lawnmower which can reflexively avoid obstacles. While LIDAR was previously used by the robot to determine obstacle locations, the sensor's price makes its inclusion in commercial versions prohibitively expensive. Cameras can provide similar information at drastically reduced cost, but useful information must first be extracted from incoming images. This can be computationally expensive. Additionally, vision-based methods can be highly sensitive to changing lighting conditions. This thesis presents a method to identify grass based on color and visual texture classifiers for use in an outdoor environment. Neighborhood-based color measurements are calculated using the HSL color model and texture measurements are based on edge-detection and quantified via computationally inexpensive first and second order statistics. Individual measurements are then combined to create a binary representation of mowable terrain in an image. Performance is quantified by measuring recognition performance on a set of sample neighborhoods that contains common backyard obstacles.

# 1. Introduction

Autonomous personal service robots are becoming increasingly popular with users around the world. After a long day of work, many people do not want to spend their precious free time performing menial chores that require little active thought and are comprised of mind-numbingly simple, repetitive actions. With society's increasing desire for efficiency and the ever-growing ubiquity of cheap technology, it makes sense to create low-cost robots for the home that are designed to perform very specific chores. Companies, such as Boston's iRobot Corporation, have recognized this, and have built successful multi-million dollar businesses around supplying consumers with little automatons that free up their owners' schedules. A veritable army of Roombas now vacuums millions of carpets every day across the globe.

But what about outside the living room? While robotic vacuums keep the carpet presentable, homeowners still woefully trudge to their garage every week to spin-up their lawnmowers and manually cut their grass. Yet, vacuuming and mowing grass are surprisingly similar chores. Both require the user to fully travel over the entire operating surface, all the while avoiding obstacles, but getting as close to them as possible. Using similar control methods as the Roomba, several companies have recently introduced robotic lawnmowers that seek to eliminate human interaction in this task as well. Unfortunately, the Roomba's method of randomly travelling around in its operating environment is ill-suited for lawn mowing, since it yields a low quality of cut and cannot guarantee that all of the lawn is mowed before the robot's batteries die.

In 2008, the CWRU Cutter (pronounced "crew cutter") team at Case Western Reserve University's Center for Biologically Inspired Robotics Research set out to create an intelligent autonomous lawnmower that could achieve the same quality of cut as a human operator. The students demonstrated that it is possible for a robot to autonomously, efficiently, and safely mow in an applied environment representative of typical suburban yards, while achieving an aesthetically pleasing end result. The robot's performance was tested during the Institute of Navigation's Autonomous Lawnmower Competition, an annual, international contest held in Dayton Ohio. In 2008, during its first year in competition, CWRU Cutter finished in 3rd place. The subsequent year, CWRU Cutter finished 1st, besting six other teams.

Unlike currently available mowing robots, which sense obstacles through direct contact, the first CWRU Cutter relied on a Light Detection and Ranging (LIDAR) sensor to safely sense obstacles from a distance and intelligently edge around them. Unfortunately LIDAR's price, greater than $5000 USD, makes the sensor's inclusion in commercial versions of CWRU Cutter prohibitively expensive. Cameras, on the other hand, can provide a plethora of information about a robot's operating environment at a fraction of the price. However, unlike LIDAR, which natively returns the range between the robot and the nearest obstacle, useful information must first be extracted from incoming images by identifying what areas in the images are safe to drive on and what areas contain obstacles.

This thesis describes a method of identifying grass in an image based on its perceived color and visual texture and segmenting incoming images based on these identifiers to differentiate between mowable terrain and obstacle locations. Chapter 2 outlines a brief history of technology in currently existing autonomous lawnmowers and highlights their differences compared to CWRU Cutter. An overview of digital image composition, color space, and visual texture follows. Chapter 3 details how grass is identified in an image based on color and visual texture identifiers and discusses how results from multiple identifiers are combined to create a robust indication of grass locations in the image. Chapter 4 discusses the methods used to quantify how functional each identifier is and presents the results to the reader. Additionally, this chapter examines the results and suggests how individual identifiers should be pre-weighted before combining them to create an accurate representation of grass containing regions in an image. Finally, Chapter 5 summarizes the results and presents the reader with possible future work that can be done to improve recognition performance.

# 2. Background

## 2.1 Robotic Consumer Lawnmowers

Since 1997, several companies have offered commercial robotic lawnmowers for global markets [1]. These robots detect obstacles in a similar manner to the Roomba – a perimeter wire is installed around permanent obstacles such as trees in the yard which inhibit obstacle collisions, and other semi-permanent obstacles, such as lawn chairs, are detected through direct contact or infrared (IR) sensors. Relying on perimeter wires and IR sensors for navigation and obstacle avoidance yields several benefits, but also has several detriments.

By physically installing borders into the robot's operating environment, it is possible to successfully mow autonomously by employing random walking. Random walking for a robotic lawnmower posits that for a finite yard space, all of the grass will eventually be cut as time approaches infinity. With perimeter wires in place, it is impossible for the robot to leave the yard or collide with obstacles specifically outlined during installation. If the direct contact or IR sensors detect an obstacle that is not outlined by the perimeter wire inside of the robot's environment, current generation robot lawnmowers will simply turn and randomly head in another direction away from the sensed obstacle. The benefits of this control strategy are that it is easy to implement and the robot needs few sensors, since it never needs to know its real-world position in the yard. This eliminates computational complexity and brings down the overall price of the robot, since it does not need to have fast onboard processing to process information from multiple sensors. However, this approach is also subject to many detriments.

First, IR sensors are not able to detect obstacles more than a half a meter away[1]. Therefore, obstacle detection through IR or direct contact presents serious safety hazards to non-outlined obstacles in the robot's mowing environment, since the robot must first come uncomfortably close to an obstacle before it realizes one is there. Additionally, IR sensors are extremely sensitive to changing lighting conditions, and can be blinded in areas of bright sunlight[1]. As such, lawnmower manufacturers warn users to clear their yards of any obstacles and maintain a safe distance from the robot while it is mowing in their product documentation manuals [3].

Since it is not possible to rely on IR sensors in an outdoor environment, the majority of lawnmower obstacle detection is accomplished via direct contact switches. By definition, direct contact is non-ranged and therefore it is virtually impossible for the robot to navigate intelligently around obstacles. It is also possible for the mower to miss large patches of grass as it randomly turns away from an obstacle, which requires the user to re-mow their lawn after the robot has finished operating. This results in a low quality of cut, since mowed lines are not parallel or in any pattern [FIGURE 1].

[1] Based on personal testing.

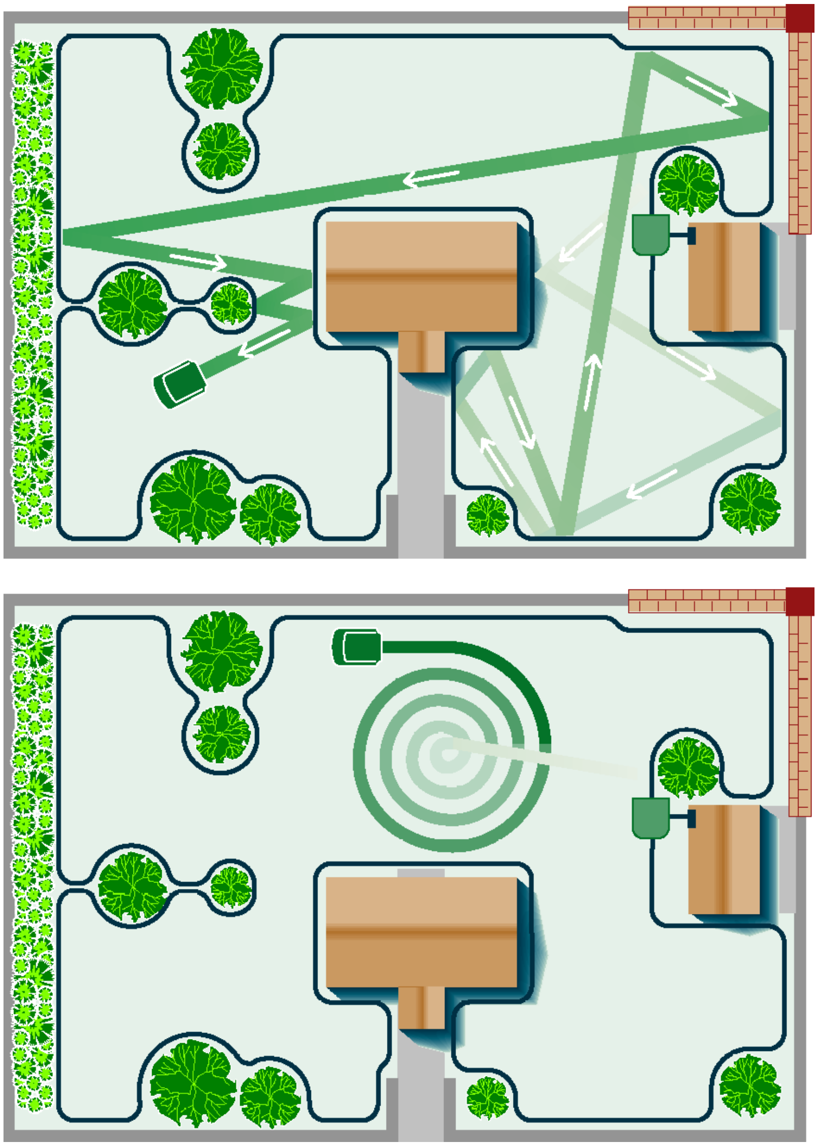


FIGURE 1: Cutting paths of currently available consumer lawnmowers. [3]

## 2.2 Autonomous Vehicles

Unlike mobile robots, autonomous vehicles are unmanned ground vehicles that aim to navigate real-world environments without human intervention by relying on sophisticated position estimation and ranged obstacle sensing techniques. While autonomous vehicle history can be traced back to the simple driverless car designed in 1978 by the Tsukaba Mechanical Engineering Laboratory that followed white visual markers in a highly structured environment, full vehicle autonomy was not demonstrated until the 2004 Defense Advanced Research Projects Agency (DARPA) Grand Challenge [4][5].

The competition tasked entrants to design and build a completely autonomous vehicle that could successfully navigate a 150 mile desert course from Barstow, California to Primm, Nevada. During the competition, vehicle localization was accomplished via GPS and a variety of other positional sensors, such as wheel encoders and inertial measurement units (IMU), while obstacle avoidance was accomplished via laser scanning [5]. Though the course would not be completed by a team until the following year's competition, the 2004 Grand Challenge defined what future autonomous vehicles needed to be able to accomplish. Instead of randomly operating within a small, highly controlled zone with some human interaction, autonomous vehicles need to intelligently follow paths in large, highly unstructured environments. Additionally, autonomous vehicles need to sense obstacles from a distance and avoid them while being able to maintain their planned route.

The Intelligent Ground Vehicle Competition (IGVC), founded by the Association for Unmanned Vehicle Systems International (AUVSI), has also championed the development of autonomous vehicle technology since the 1990's [6]. While the IGVC course is not as massive as the Grand Challenge's, the competition tasks university level engineering students to autonomously navigate a marked off-road course while performing increasingly complex tasks along the way [6]. Oftentimes, IGVC vehicles supplement laser-based ranging with computer vision to successfully identify obstacle and navigate around them [7].

The autonomous lawnmower, CWRU Cutter, represents a convergence between consumer robotic lawnmower application and autonomous vehicles technology to create an intelligent, autonomous robot that is able to sense obstacles from a distance and mow in straight lines, resulting in a lawn with a comparable quality of cut that a human operator can achieve.

## 2.3 CWRU Cutter

CWRU Cutter (pronounced "crew cutter") is an autonomous lawnmower designed and built by students at Case Western Reserve University's "Center for Biologically Inspired Robotics Research [FIGURE 2]." The robot is an off-shoot of CWRU's involvement in the 2007 DARPA Urban Challenge and several IGVCs, and is more related to the university's autonomous vehicle approaches than to currently available commercial robotic lawnmowers. Yet, CWRU Cutter addresses consumer desire to own an autonomous lawnmower. In this sense, the robot is meant to improve on currently existing, commercially available autonomous lawnmowers by creating a system that is able to navigate intelligently through a yard and achieve a quality of cut comparable to a human operator.

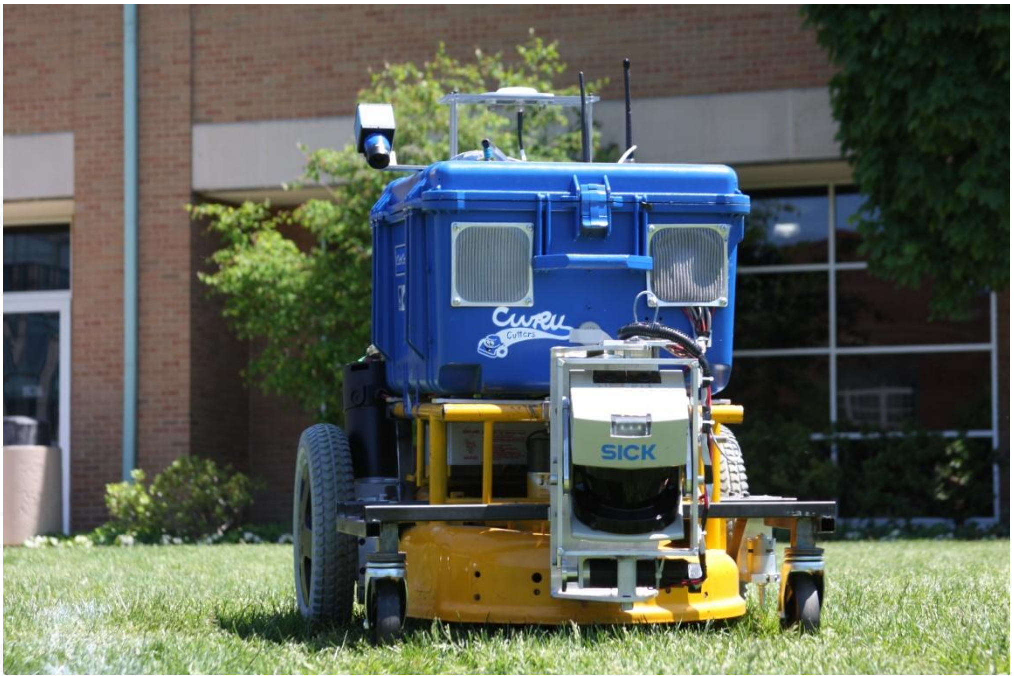


FIGURE 2: The 2009 CWRU Cutter Autonomous Lawnmower

However, mowing yards is not a trivial task for robots. Unlike the DARPA challenges, where courses are clearly defined in terms of GPS waypoints, or the IGVC, where the robot's path is bounded by white lines, CWRU Cutter must function in an area where no clear physical boundary markers exist and obstacles litter its operating environment. The Institute of Navigation's (ION) "Autonomous Lawnmower Competition" is designed to be representative of a typical suburban environment and highlights many of the challenges that would be faced by commercial autonomous lawnmowers [8].

In the competition, robots are placed in an irregularly shaped course, which contains obstacles such as white fence and flowers with flowerbed edging that are semi-randomly placed and not outlined before the competition run. To achieve a high score, the robot must carefully edge along these obstacles, cutting as much grass along their border without damaging them. The robot must also avoid a moving obstacle, a stuffed dog mounted on a remote controlled car, and execute ranged sensing and avoidance behavior as the dog passes in front of the robot [8].

To accomplish these tasks, CWRU Cutter relies primarily on differential GPS for positional information and, in previous years, relied on a "light detection and ranging (LIDAR)" unit to determine relative distances between itself and the nearest obstacle [9]. LIDAR natively outputs ranges in a polar coordinate system between the (0,0) location of the unit, and the distance, in millimeters, to the nearest obstacle in front of the sensor within a 45º angular range relative to the sensor's forward facing plane at 0.5º angular increments [FIGURE 3].

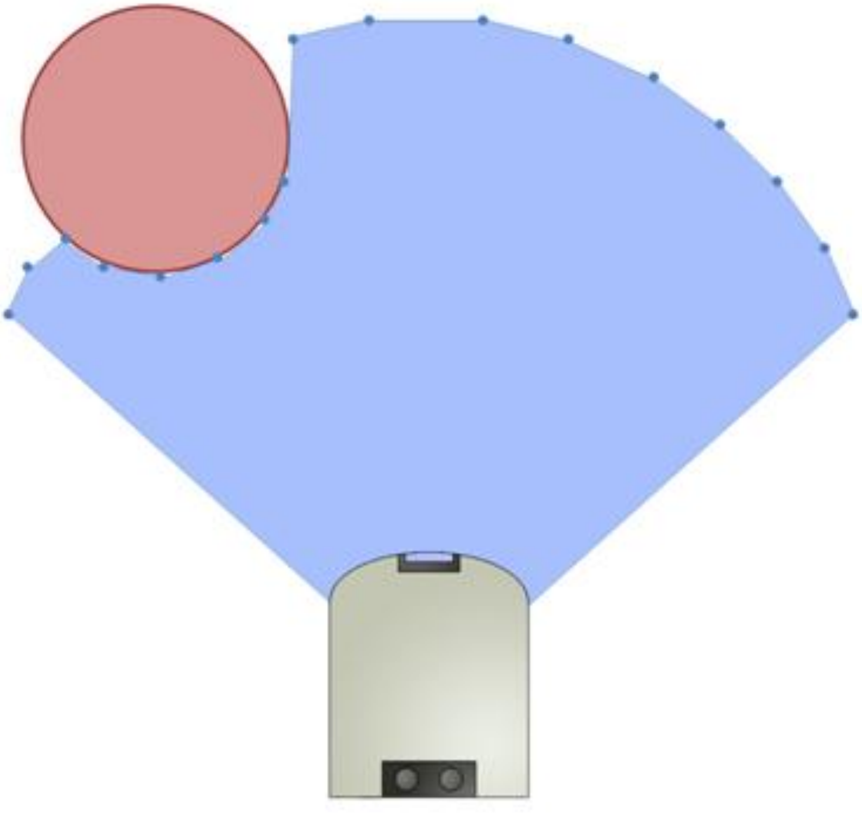


FIGURE 3: An illustration of a LIDAR scan. The LIDAR unit (bottom) sends out laser readings at 0.5º angular increments (blue dots), which reflect off of obstacles (red circle). This returns ranges that are used to calculate amount of space between the sensor and the nearest obstacles (light blue area). If an obstacle is not observed, the laser is not reflected and a range equal to the maximum observable distance for the sensor is recorded for that angle.

As the robot drives through its environment, it remembers the previously observed freespace immediately behind it, and subsequent LIDAR readings that measure ranges to the nearest obstacles in front of the robot are combined via a Kalman filter. Combining what the robot remembers with new LIDAR readings creates a 360º description of ranges to the closest obstacles immediately surrounding the mower [FIGURE 4].

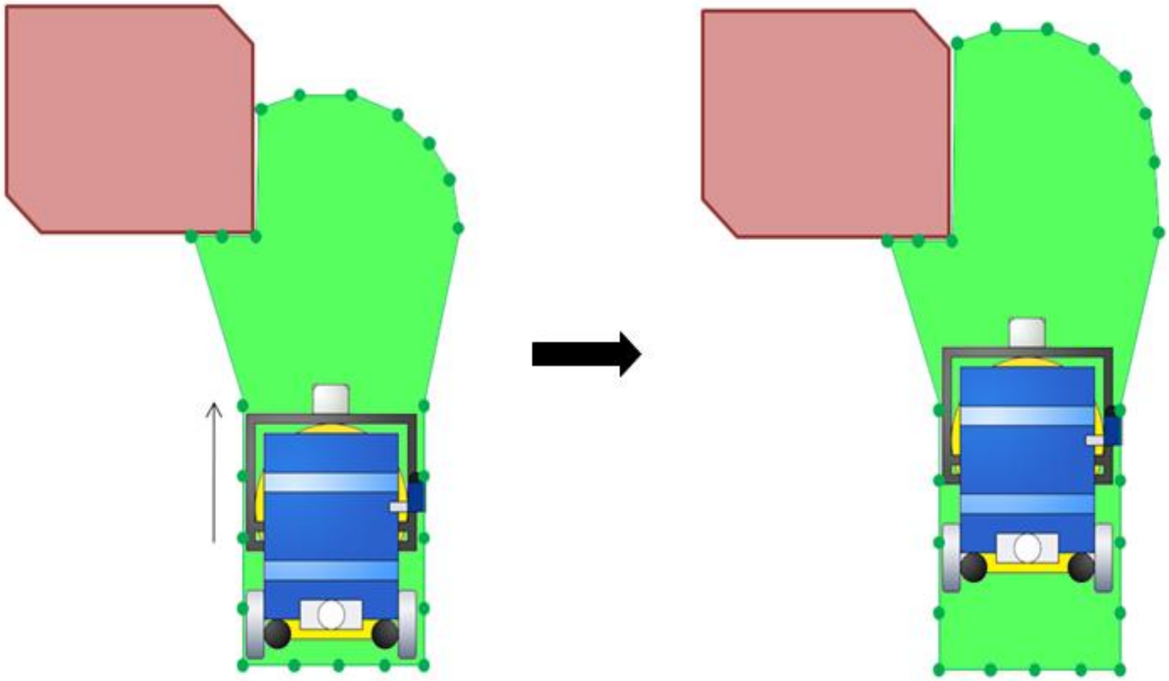


FIGURE 4: An illustration of polar freespace. The lawnmower (bottom) approaches an obstacle (red shape), while the LIDAR is continuously measuring the amount of space between the robot and obstacle locations within the sensor's field of view (green shape). Space currently occupied by the robot is also considered to be obstacle free. This creates a 360º of space around the robot that the mower can travel to. As the robot travels forward, it remembers previously observed freespace and only updates the ranges in front of the mower that the LIDAR directly observes (right). The 360º representation of space that the mower can travel to is referred to as the polar freespace.

Tracking the freespace immediately around the robot allows it to perform ranged detection and reflexive avoidance of obstacles in the yard without needing to lay down a perimeter wire. During operation, the robot follows a sequence of GPS waypoints, which define its mowing path. GPS waypoints are automatically generated sequential locations within the yard that the mower must traverse while it is mowing. As the robot senses an obstacle in its generated mowing path, it will gradually slow down as the measured range between it and the observed obstacle decreases.  Once the robot gets close enough to the obstacle, it will turn to avoid it.  The robot then drives along the obstacle, continually attempting to turn into it in an attempt to travel to the nearest GPS waypoint and resume its mowing path.  This creates a reflexive edging behavior that allows the robot to cut along all observed obstacles in its mowing environment in a consistent way.  If the mower passes a GPS waypoint during this reflexive edging behavior, it will look towards the next GPS waypoint in its mowing path, and try to reach it.  Once an obstacle has been passed, it will resume its pre-generated mowing path and follow GPS waypoints until it encounters another obstacle and repeats this behavior [FIGURE 5].

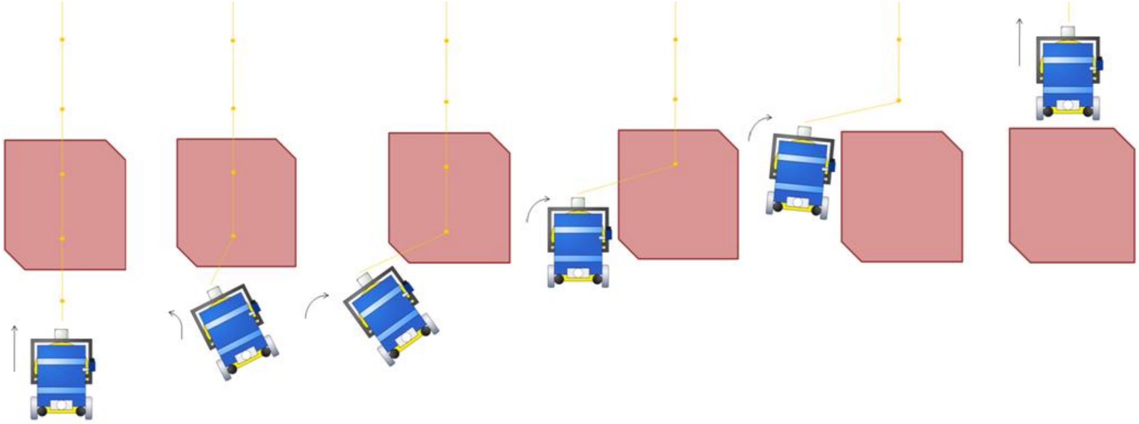


FIGURE 5: An illustration of reflexive obstacle avoidance.  The mower's driving behavior in each image is indicated by the black arrow.  (Far left) The lawnmower (bottom) encounters an obstacle (red shape) while following its mowing path (yellow line) generated by connecting consecutive GPS waypoints (yellow dots).  As the robot gets close to the obstacle, LIDAR observes it and the robot begins to turn left to avoid the obstacle.  While moving forward, it continuously attempts to turn right into the obstacle in an attempt to travel to the nearest GPS waypoint, but cannot, because LIDAR continues to observe that the obstacle is present.  Eventually, the robot passes a GPS point.  It then looks towards the nearest waypoint and continues to turn in an attempt to reach it.  It again does not reach it before passing it and moves on to the next waypoint.  Since this waypoint is not located inside of the obstacle, it turns right, reaches it, and continues mowing along the same line as it did before encountering the obstacle.

By coupling GPS positional information with generated polar freespace measurements, the robot achieves a high quality of cut, since it follows a structured path and mowing behavior, while being governed by dynamic, unplanned obstacle avoidance rules. While CWRU Cutter's current obstacle detection and ranging system, which is based solely on LIDAR, is robust and appropriate for a proof-of-concept robot, it has several major drawbacks. First, it is only able to observe obstacle locations and cannot differentiate between obstacle types. Therefore, it will identify a tall blade of grass and flowerbed edging or feet identically. Since LIDAR can only observe obstacles in one plane, it will not see obstacles that are outside of its field of view. In the 2008 CWRU Cutter, this presented a problem for mowing, since the sensor orientation was fixed on the robot, and the sensor could not see low obstacles, such as ground cover. For a consumer robot, LIDAR had one additional major drawback – price.

While a LIDAR is an extremely robust sensor and is capable of functioning in a wide variety of environments and lighting conditions, even low-end models cost upwards of $5000 USD, and are therefore prohibitively expensive for inclusion in a commercial autonomous robot that consumers can purchase at their local hardware store [11]. To retain functionality of CWRU Cutter's current obstacle avoidance system, it is therefore necessary to replace LIDAR with a sensor that can output the same information less expensively. One such sensor is a digital camera.

## 2.4 Cameras & Computer Vision

With their rising ubiquity in almost all areas of autonomous industrial systems and consumer electronics, digital cameras are available at drastically lower prices than LIDAR. Though cameras do not natively output data in inherently useable formats like a LIDAR, incoming images can easily be abstracted into multiple forms, which make them extremely versatile. Though image abstraction methods are oftentimes computationally intensive, computer visions systems are now widely used in autonomous robotic applications, thanks to ever increasing processor speeds [12][13][14]. The computational complexity issue for real-time computer vision applications, however, is non-trivial and must always be balanced with the computer-vision field's primary interest – how to identify useful information in the image to abstract in the first place.

Since computer vision is so versatile, strict methods of how images are analyzed do not exist, and, as such, applications for vision-based sensing are highly varied. Many DARPA Challenge robots have utilized stereo-vision for ranging and car identification that was integrated with LIDAR information to govern lane-changing behavior [15]. During the 2005 Grand Challenge, Stanford University's autonomous vehicle "Stanley" relied on road color information to control the vehicle's driving speed [12]. Computer vision's versatility has led researchers to extract useful image information in multiple ways, the most successful of which have been color and visual texture.

### 2.5.1 Color

Digital images are composed of 2D arrays of pixels [16]. The number of pixels in an image is given by its resolution. For example, a 640x480 pixel image contains 640*480 (307,200) pixels. A pixel can assume a color value, which can be defined using various color models.

One of the earliest and most widely used color models is "Red, Green, Blue (RGB)" [17]. The theory of the RGB color model is based on the Young–Helmholtz theory of trichromatic color vision, which was developed in 1850, and mirrors the three types of color receptors in the human retina [17]. The color model contains three primary colors – red, green, and blue. The model is additive, and states that when these three primary colors are mixed together in different ratios, it is possible to create a multitude of colors, from black, which is created when none of the colors are present and is represented by the RGB vector <0,0,0> to white, which is created when all of the colors are present in the fullest amount and is represented by the RGB vector <255, 255, 255> when utilizing a 24-bit color model [16] [FIGURE 6]. However, RGB is not the only color model which can be used to represent a color.

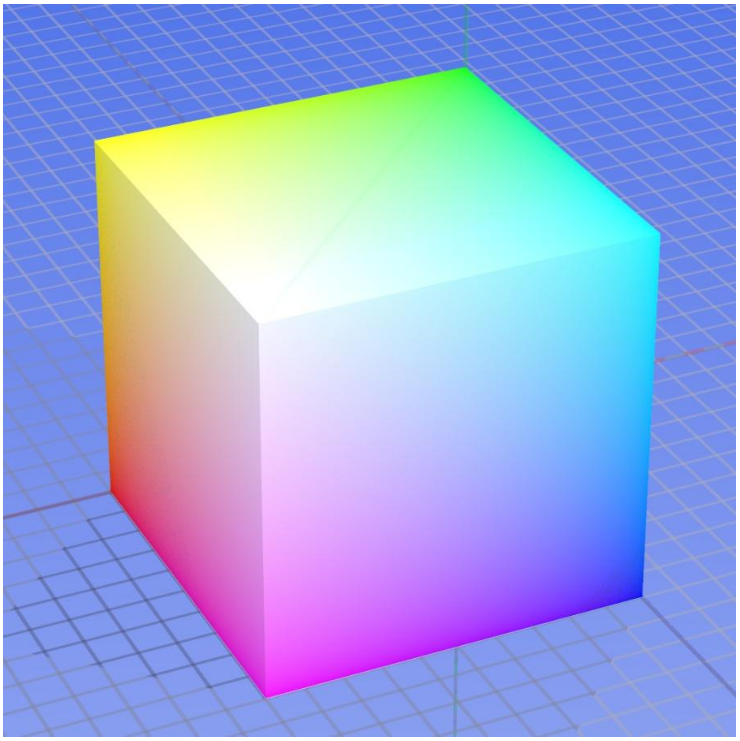


FIGURE 6: The RGB Color Cube [GNU Image: Wikipedia]

In the 1970s, Xerox and the New York Institute of Technology developed the HSV color model as an alternative to RGB for inexperienced users to describe color based on human intuition [18]. HSV stands for "Hue, Saturation, and Value." During this time, a similar color model called "HSL/HSI" was also developed, which replaces the "value" component with a "luminance/intensity" component [18]. In these models, the hue component describes the base color of a pixel. Saturation describes the amount of the present color, and luminance describes how light or dark the pixel's color is. For example, let us intuitively describe the color of a coffee cup [FIGURE 7 left]. Observing the cup, we say that it is blue (hue). However, relative to other blues we perceive, such as a Blu-ray disc case, we say that the coffee cup is *very* blue (saturation) [FIGURE 7 right]. Finally, the blue observed in the coffee cup is relatively dark compared to the Blu-ray disc case (luminance/intensity). By quantifying these descriptions numerically, it is possible to create a representation of the HSI color space [FIGURE 8].

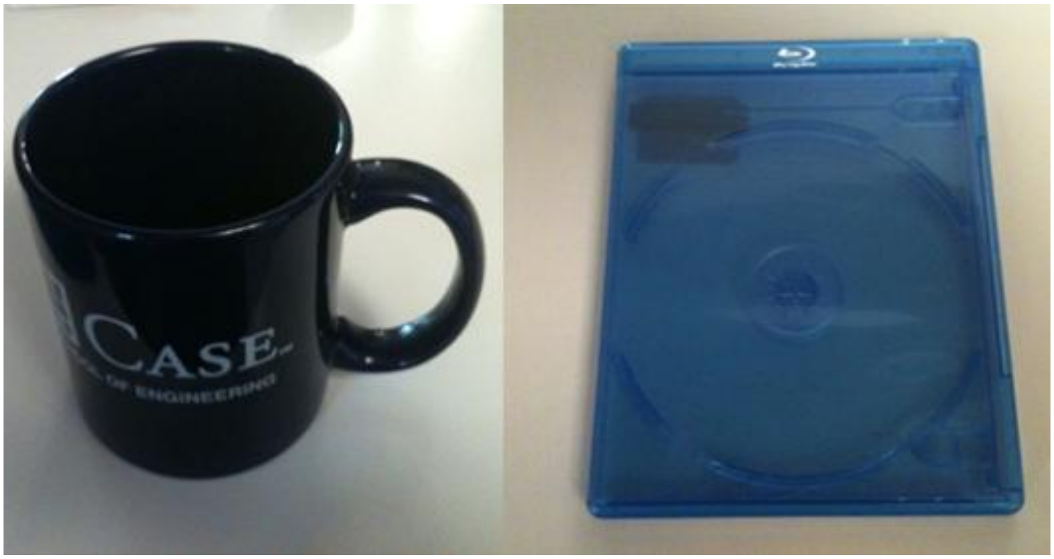


FIGURE 7: Hue examples. CWRU coffee cup (left) and a Blu-ray disc case

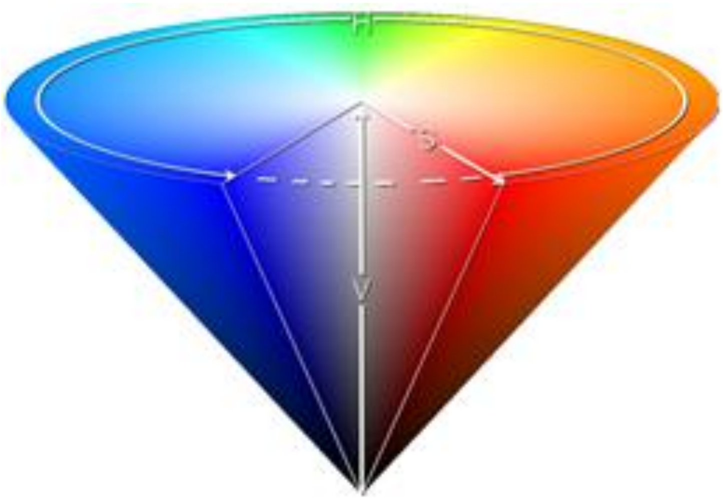


FIGURE 8: The HSL color cone [19]

It is possible to convert between RGB and HSI color representations by calculating the individual components of the new color model based on the available model's components.

### 2.5.1.1 Color Applications for Autonomous Vehicles

Since a pixel's color vector is easily read and does not require complex computations to extract, color usage has been extensively investigated for application in autonomous vehicles to identify objects of interest and to determine drivable terrain [12][13].

Object identification via RGB functioned satisfactorily for identifying geometrically simple objects with constant color under highly controlled lighting conditions or in an outdoor environment where lighting conditions were fairly constant [20][12]. However, RGB representation is exceedingly sensitive to changing lighting conditions, since a color under different illumination is represented by a completely different color vector [20]. Therefore, its applications in outdoor environments for mobile robot navigation are limited.

HSI on the other hand was found to be insensitive to changing lighting conditions, since the hue component remained relatively constant, as the underlying object's color did not change when the object was occluded by shadows [21][20]. As such, HSI has been widely applied in agricultural applications for autonomous weed identification and spraying [22][23]. It is important to note that while HSI color representation is more robust than RGB, identification failures occur when identifying objects with high color variation or objects under extreme lighting conditions. Therefore, it is necessary to rely on additional visual quantifiers to improve recognition performance. One such measurement is visual texture.

### 2.5.2 Texture

With recent increases in computational power, visual texture has become a way of robustly identifying targets of interest in an image for autonomous vehicle applications. Texture aims to quantify underlying target surface characteristics that are color and illumination independent. For example, the texture of cardboard would intuitively be described as "smooth," whereas carpet is "bumpy" and "less smooth than cardboard," and grass is "messy" [FIGURE 9]. To identify texture based on these qualitative descriptors, it is necessary to extract and quantify visual texture of various objects. Unfortunately, unlike image color, which is clearly defined in computing through widely accepted color

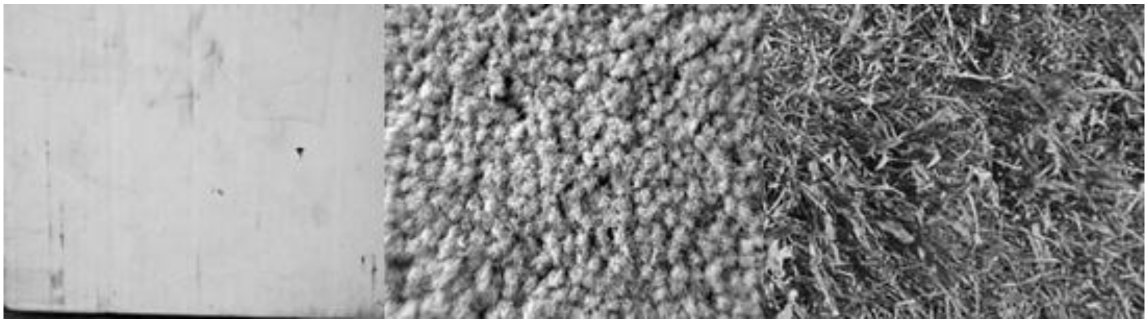


FIGURE 9: Surfaces with various visual texture. (From left to right) cardboard, carpet, and grass.

models, researchers use various texture descriptors depending on what best suits their application [16][24]. Therefore, no universal descriptors of what constitutes “texture” exist.

One of the more intuitive methods of quantifying visual texture is through edge detection. Edge detection is an image processing technique which aims to detect boundaries based on color discontinuities between a center pixel and its surrounding neighbors across one component of the color vector in an image [25]. Like components of a color vector for an entire image are known as a “color plane” and pixel areas within an image are known as “pixel neighborhoods.” Discontinuities are quantified by calculating an oriented gradient within the neighborhood and plotting the magnitude of the result at the center pixel [FIGURE 10]. Larger magnitudes correspond to larger discontinuities, which result in a stronger edge response.

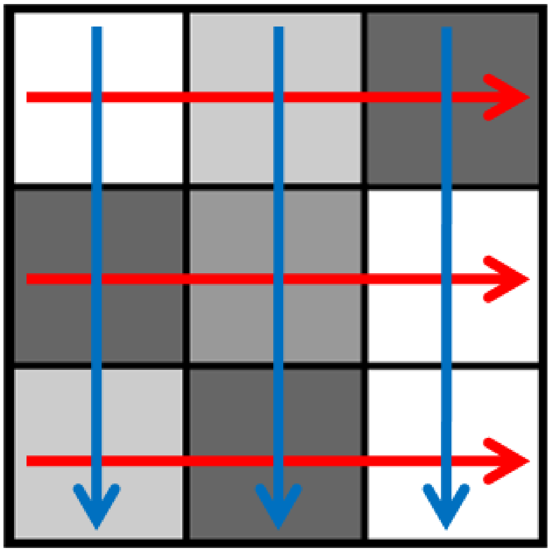


FIGURE 10: Illustration of oriented gradient used to calculate edges in a 3x3 pixel neighborhood. Red arrows indicate calculation of horizontal edges and blue arrows indication calculation of vertical edges. The magnitude of both differences is plotted at the center pixel and indicates the edge response strength for that neighborhood.

Edge detection is a local descriptor, since edge magnitudes are based on relative values between a pixel and its neighbors. This is an important quality for visual texture, since an illuminated neighborhood will produce the same edge response as a shaded neighborhood, due to the fact that the *relative* color changes between the pixels is the same for both neighborhoods [FIGURE 11].

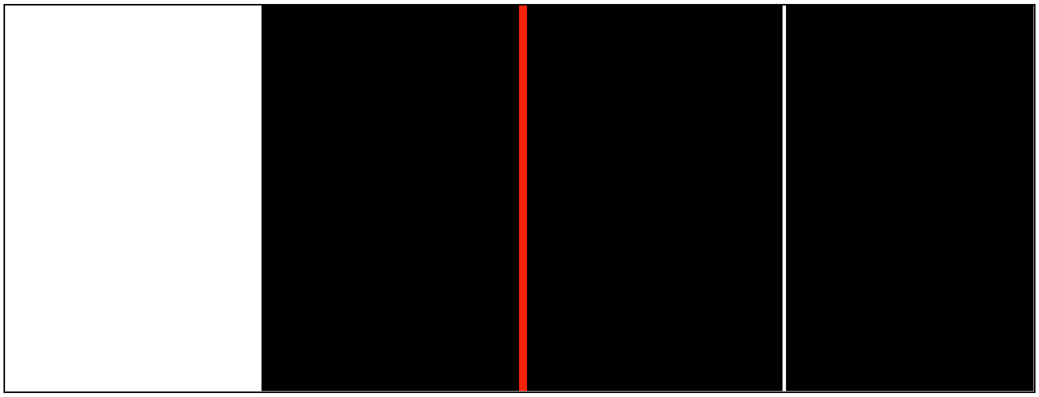


FIGURE 11: An example of edge detection. The original image (left) and the resulting image (right) after edge detection. Images are separated by the red line. A horizontal edge detector is applied to the left image, where no color change occurs until the middle of the image, where color changes the maximum possible amount in RGB space, from 0 to 255. This results in a strong edge response in the middle three pixels of the image, indicated by a vertical white line.

Gradient neighborhood calculations are performed in computing via convolution. To perform a convolution, it is necessary to define a convolution kernel. A convolution kernel is a square matrix, whose dimensions correspond to the desired neighborhood size a user wishes to calculate the gradient for and the elements in the matrix correspond to weights of the individual rows or columns in the magnitude calculation. A common convolution kernel for edge detection is a Prewitt kernel, which calculates the equally weighted horizontal or vertical derivates in a pixel neighborhood. For a 3x3 neighborhood, the horizontal & vertical Prewitt filters (*Gx* and *Gy*, respectively) are given by:

$$Gx = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \qquad (1)$$

and

$$Gy = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}. \qquad (2)$$

Filter directionality is also an important property of image convolution, since oriented filters will respond differently to oriented edges in an image. For example, convolving Figure 11 with *Gx* yields a strong edge response, since the convolution encounters a vertical line with a stark color difference in the middle of the image. Convolving *Gy* with the same image yields no edge response, since there is no color change between neighboring vertical pixels in the image [FIGURE 12].

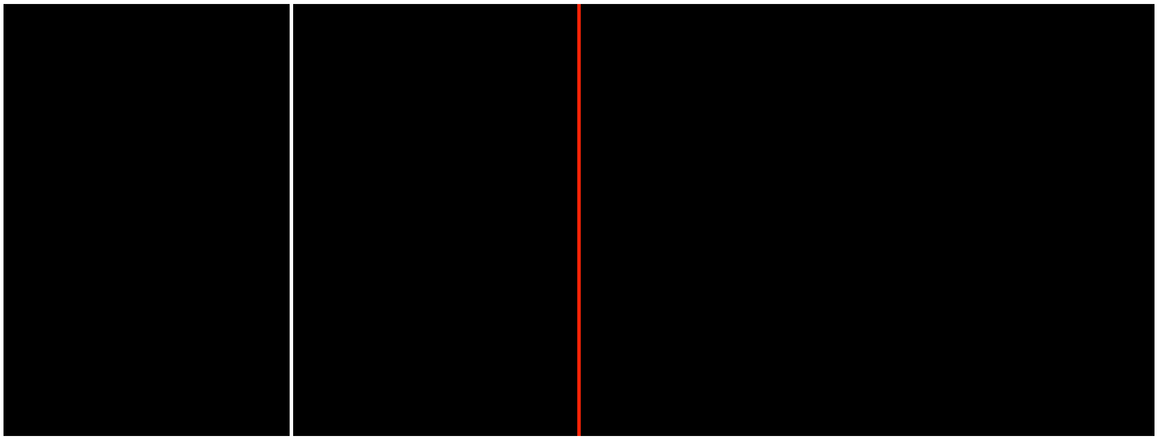


FIGURE 12: Edge detection outputs for horizontally oriented edge detector (left) and vertical edge detector (right) for the original image presented in FIGURE 11. In this example, Images are separated by the red line. While the horizontal detector is able to find an edge, no edges are detected by the vertical detector, as the colors in the columns of the image are constant.

Edge detection makes underlying surface characteristics more apparent. Applying *Gx* to Figure 9 creates a color independent description of the samples' visual textures [FIGURE 13]. It is now necessary to quantify these textures numerically, so that they can be identified by a computer. A common method is identification via "block statistics" [26]. In this method, a collection of edges within a pixel neighborhood is defined as the visual texture. Statistics of the edge response magnitudes for each pixel within the neighborhood are then calculated. This yields statistical values that describe various visual textures which are then used to identify and segment surfaces in incoming images.

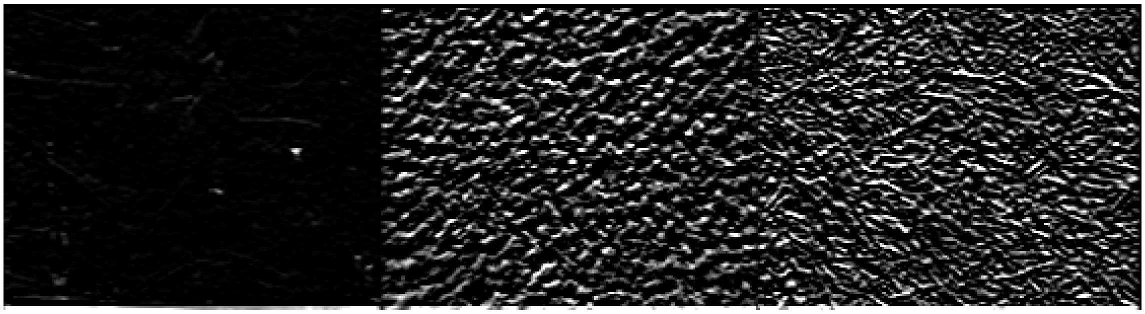


FIGURE 13: Horizontal edge detection results of Figure 9. Each sample has an intuitively apparent, distinct visual texture.

### 2.5.2.1 Texture Applications for Autonomous Vehicles

Since edge-based texture is a relative measurement based on neighboring pixels, it functions well in environments where lighting conditions are not strictly controlled. This makes it an ideal method of identifying objects in outdoor environments, where surfaces can exhibit large color variations and inconsistencies due to solar illumination, or where the color is unperceivable due to shadow occlusion.

Various visual texture measurements have been successfully utilized for robot navigation on autonomous vehicles to determine drivable terrain around the robot or to identify objects of interest [26][27][28]. Texture's application for autonomous navigation can depend largely on how the sensor is mounted on the robot. As the camera is mounted at a shallower angle relative to the vertical axis on the robot, it can perceive objects further away. The distance between the camera and a perceived object is referred to as the "camera-object distance." Since cameras have fixed resolutions, they perceive less texture detail about objects that are further away. Perceiving less detail acts as a natural low pass filter and images of distant objects will have less variation in surface texture that is due to small objects like ground cover, weeds, or pebbles. When the camera can see far into the distance, visual texture is well suited for looking at textures over a large area, which makes it an ideal method of identifying drivable terrain.

During the 2007 DARPA Urban Challenge, multiple robots used visual texture to determine road surfaces [27][28]. In these applications, the cameras were mounted at least a meter above the ground and had a camera-object distance on the order of 10-100m. Therefore, cameras on these robots were able to perceive large scale road texture to keep the robots on course.

Conversely, visual texture can also be used to identify objects of interest when the camera is mounted closer to the ground and has a small camera-object distance. In 2006, Watchareeruetai et al. demonstrated a texture-based method of identifying weeds for targeted herbicide spraying on an autonomous lawn care vehicle [26]. In this case, the camera-object distance was on the order of 1 meter. Utilizing this method, the researchers were able to achieve a 91.1% successful weed identification rate and demonstrated that visual texture could be used to extract information about small objects in an image when the camera geometry was set up appropriately. The closer the camera is to the object of interest, the more pixels are dedicated to smaller real-world space. Therefore, it is also possible to extract minute details about individual objects using texture.

These methods serve as motivation for a visual-based obstacle detection and navigation system for CWRU Cutter.

# 3. Methods of Grass Identification

## 3.1 Motivation

The goal of this research is to create a robust method of identifying grass in an image based on color and edge-based visual texture and to quantify their descriptors. This information will be used by CWRU Cutter for reflexive obstacle avoidance. The future goal of the project is to replace the currently used LIDAR sensor with an array of cameras, thereby decreasing the robot's overall price and increasing its commercialization potential.

Due to the inherently destructive nature of a lawnmower, it is unacceptable for the robot to collide with any obstacles during operation. Therefore, all obstacles the mower encounters are considered equally important. By identifying grass in an image, the mower intrinsically knows all obstacle locations in its current field-of-view, since anything that is not grass is an obstacle. The complexity of creating a vision-based obstacle identification system becomes much simpler when the problem is framed this way, since it exploits a key detail about lawn mowing – the robot should only be cutting while it is moving on grass.[2] Due to the variability of objects the robot could encounter while it is mowing, classifying and training the robot to recognize every known obstacle is an impossible task. However, if the robot can successfully differentiate between grass and everything else with a high degree of accuracy, it is possible to create a low-cost,

[2] Though a lawnmower should only *mow* grass, this is not necessarily the only surface a lawnmower drives over during operation (for example: turning around on the sidewalk). The set of "drivable" terrains was restricted to grass for simplicity. If grass can be successfully identified by a range of statistical values utilizing the proposed measurements, it is likely that other, visually distinct surfaces can be described by different ranges for the same statistic.

vision-based, safe autonomous mowing system that can be used in future, commercial versions of CWRU Cutter.

### 3.2.1 Grass Identification via Hue

Image hue is used to identify grass in the image based on statistical measures that are calculated from randomly selected 50x50 pixel neighborhoods extracted from randomly selected images of grass in the collected data set where each neighborhood contains only either illuminated or shaded grass. The following objects and number of neighborhoods were used for statistical calculation – 40 neighborhoods of illuminated grass and 10 neighborhoods of shaded grass.

Since the raw camera images are represented in the RGB color space, it is first necessary to calculate the hue component of the HSI color vector according to:

$$\begin{aligned} h &= \cos^{-1}\left(\frac{0.5\left[(r-g)+(r-b)\right]}{\left[(r-g)^2+(r-b)(g-b)\right]^{0.5}}\right) \quad h \in [0,\pi] \quad \text{for } b \le g \\ h &= 2\pi - \cos^{-1}\left(\frac{0.5\left[(r-g)+(r-b)\right]}{\left[(r-g)^2+(r-b)(g-b)\right]^{0.5}}\right) \quad h \in [\pi,2\pi] \quad \text{for } b > g \end{aligned} \tag{3}$$

where *h* is the normalized hue component and *r*, *g*, and *b* are the normalized components of the original RGB vector [24]. The normalized RGB components are given by

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

$$g = \frac{G}{R+G+B} \tag{5}$$

and

$$b = \frac{B}{R+G+B}\,. \tag{6}$$

Since HSI color space can be represented by a cone where all possible hue values lie on the cone's plane base, hue values can be represented by an integer in the range [0, 360). These values are represented as integers due to the way colors are represented in digital images. Since (3) is the normalized hue component, it is necessary to scale the normalized hue component *h* into this range through the equation

$$H = (\mathrm{int})\left(\frac{180h}{\pi}\right) \tag{7}$$

where (int) represents a typecasting operation of the scaled value into an integer and *H* is the scaled hue component of the HSI color vector.

It is important to note that singularities occur in (3) when *r*, *g*, and *b* lie along the RGB color cube's grayline and therefore have the same numerical value. This results from color representation in the HSI color model where white, black, and grays have an undefined hue.

Once the hue component is computed for each training neighborhood, the mean and standard deviation of the hue for the neighborhood is calculated. The average of the mean neighborhood hue values is then computed across the set of neighborhoods.

The hue is calculated for all images acquired by the robot and all pixel hues within two standard deviations of the previously calculated mean hue are marked as grass. Pixels that fall outside of this range are marked as obstacle containing. If more than 50% of the pixels within a neighborhood are identified as grass, the entire neighborhood is marked as grass containing. This creates a binary (grass/not grass) representation of drivable terrain in the image, which can be abstracted to generate a polar freespace representation around the robot as described in Chapter 2.

### 3.2.2 Grass Identification via Texture

A combination of seven edge-based texture statistics are used to identify grass. These statistics are calculated for non-overlapping pixel neighborhoods in an approach known as "block statistics" [26]. Similar to the hue processing described in the previous section, statistical measures used to identify grass and objects in the image are calculated from randomly selected 50x50 pixel neighborhoods from the data set. The following objects and number of neighborhoods were used to calculate these statistical measures – 40 neighborhoods of illuminated grass, 10 neighborhoods of shaded grass, 10 neighborhoods of flowers, 10 neighborhoods of white plastic fence, 10 neighborhoods of black plastic flowerbed edging, 10 neighborhoods of blue jeans, and 10 neighborhoods of a soccer ball. Since the fence, flowerbed edging, jeans, and soccer ball are man-made and do not naturally occur in nature, these objects are collectively referred to as "artificial obstacles" and their texture is referred to as "artificial texture."

The edge-based texture statistics are computed from the intensity component of the HSI image. As previously discussed, the RGB color model represents object color poorly in changing lighting conditions [21]. HSI representation, however, is more illumination independent and therefore represents object color better under various lighting conditions. To mitigate the effects of variations in lighting conditions in incoming images, edge detection is performed on the intensity component, *I*, of the HSI color model where $I$ is given by the mean value of the RGB components for individual pixels, or

$$I = \frac{R + G + B}{3}. \tag{8}$$

Horizontal and vertical edge responses are calculated from the resulting intensity image using the horizontal and vertical Prewitt convolution kernels given by (1) and (2), respectively. Each convolution yields a new grayscale image, wherein the pixel intensity given by an integer value between [0,255] indicates the edge strength at that pixel location. These edge images are referred to as "horizontal" and "vertical texture images," respectively.

These horizontal and vertical grayscale texture images were converted to binary images by discarding edge strengths below an empirically determined threshold of 21 and setting corresponding pixel locations of edge strength above the threshold to 1. Unconnected pixels in the horizontal and vertical binary texture images were removed. The horizontal and vertical binary texture images were then summed to create a directionally insensitive binary texture image corresponding to the input image. Figure 14 illustrates an example neighborhood and the resulting grayscale and binary texture [FIGURE 14].

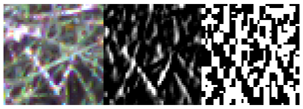


FIGURE 14: Extracted visual texture. (Left to right) RGB neighborhood, resulting horizontal grayscale texture, and thresholded horizontal binary texture.

Seven texture statistics were calculated for each sample neighborhood: 1-2.) binary horizontal and vertical variance, $Var(R)$, 3-4.) mean grayscale horizontal and vertical intensity, $\bar{I}$, 5-6.) mean binary horizontal and vertical edge response area, $\bar{A}$, and

7.) horizontal and vertical neighborhood centroid location of directionally invariant binary texture images, $(\bar{x}, \bar{y})$, respectively.

The use of variance as an important texture classifier was based on [26]. The other texture measurements were chosen for their computational simplicity, which is important for operating in real-time on a mobile robotic platform with limited hardware, as well as their relative uniqueness compared to other measurements used. Both the binary and centroid area are based on image moments (described below), which uses common elements to compute statistics. Mean intensity is merely an average of all the pixels in a neighborhood.

The variance, *Var(R)*, of the horizontal and vertical binary texture images is computed by

$$Var(R) = \sum X - \overline{X} \Big/ (n-1) \tag{9}$$

where $X$ is the binary value of the current pixel, $\overline{X}$ is the mean value of the pixels within the square neighborhood, and $n$ is the area of the neighborhood given by the total number of pixels within the neighborhood. Since we are computing the variance of a binary neighborhood, this reduces to

$$Var(R) = \frac{w - {w^2}\big/{n}}{(n-1)} \tag{10}$$

where $w$ is the number of pixels in the neighborhood above the binary threshold.

The mean grayscale intensity value of a neighborhood, $\bar{I}$, is computed by summing all pixel values of the grayscale texture images and dividing by the total number of pixels within the neighborhood.

The binary area and centroid locations of the pixel neighborhoods were calculated as

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y) \tag{11}$$

where the calculated moment $M_{ij}$ is given by the summation over the *x* and *y* dimensions of the pixel neighborhood, and *i* and *j* correspond to the index of the calculated moment. The binary area is given by $M_{00}$, and the horizontal and vertical centroid locations within the neighborhood, $\bar{x}$ and $\bar{y}$, respectively, of the directionally ambiguous texture image are given by

$$\bar{x} = {}^{M_{10}}\!\big/_{M_{00}} \tag{12}$$

and

$$\bar{y} = {}^{M_{01}}\!\big/_{M_{00}} \,. \tag{13}$$

After determining the statistical values for grass in the set of training images, corresponding texture measurements were averaged over the samples to compute the statistic's mean and standard deviation. These averages were then used to perform non-overlapping binary segmentation of new input images. For each new input neighborhood, if the measured statistic of the input neighborhood fell within an empirically determined range of three standard deviations of the previously calculated statistical measurement, that neighborhood was marked as grass. Conversely, if the neighborhood measurements fell outside this range, that neighborhood was marked as containing an obstacle. This creates a binary representation of drivable terrain based on texture in the image, which can be abstracted to generate a polar freespace representation around the robot. Individual binary representations can then be combined across the different binary drivable terrain representations for corresponding neighborhoods to potentially create an improved texture-based representation of traversable terrain in the image.

### 3.2.3 Hue & Texture Combination

While some image measures exhibit better performance than others for recognizing grass, no single measure was able to differentiate between grass and obstacles with a 100% success rate. To improve performance, individual measures were combined to generate a more accurate representation of mowable terrain in the image.

The performance for each measure was estimated by its ability to correctly classify 40 sample neighborhoods each of illuminated grass, shaded grass, and obstacles. These estimates were then used to generate a weighting coefficient for individual measurements

$$\alpha = \frac{(C_{illum} + C_{shaded} + C_{obs})}{120} \tag{14}$$

where $\alpha$ is the normalized weighting coefficient and $C_{illum}$, $C_{shaded}$, and $C_{obs}$ are the number of correctly identified neighborhoods of illuminated grass, shaded grass, and obstacles, respectively. Each weighting coefficient is normalized by dividing by the total number of image measurements (40x3=120 in this case) to generate a normalized weighting coefficient between [0,1].

Segmented binary texture images are multiplied pixel by pixel by their normalized weighting coefficients. Then, corresponding neighborhoods are added with their applied weights to generate a grayscale representation of mowable terrain in the image, known as a probability map. The likelihood of a neighborhood containing grass is determined by the sum of the weighted images across the utilized measurements. For example, if a binary neighborhood is identified as grass containing for a texture measurement and the measurement's normalized weighting coefficient is 0.1, the likelihood that that neighborhood contains grass based on that texture measurement increases by 0.1. This is repeated for all texture measures and corresponding neighborhood values are added. The sum of these neighborhoods therefore indicates the likelihood that this image region contains grass.

Let us assume that five texture measurements are used to identify grass, where four are equally weighted and one has a weight twice that of the others. The measures' normalized weights are identified as *a* through *e*, and have the following values: *a* through *d* = 1/6, *e* = 1/3. Now, assume that the texture measurements weighted by *a*, *c*, and *e* identify their corresponding neighborhoods as grass, indicated by a binary value of 1. Neighborhoods *b* and *d* state that the neighborhood is not grass, indicated by a 0. Therefore the likelihood, *L*, of that neighborhood being grass is

$$L = 1 * \left( \frac{1}{6} + \frac{1}{6} + \frac{1}{3} \right) + 0 * \left( \frac{1}{6} + \frac{1}{6} \right) = \frac{2}{3}. \tag{15}$$

# 4. Results

## 4.1 Datasets Used

All statistics were calculated from randomly selected neighborhoods that were randomly selected from images in two data sets, taken between 9:00 AM and 11:59 AM on 8/31/2009. All neighborhoods contained only one type of surface. For example: neighborhoods containing illuminated grass contained only illuminated grass, shaded grass contained only shaded grass, and plastic fence neighborhoods contained only plastic fencing. Images from both sets were recorded on MTD's Test Plot 1, which contained a mixture of rye, blue, creepy red fescue grasses[3]. These data sets were chosen for analysis for several reasons.

First, this grass mixture corresponds to the most common type of lawn grass found in Northeast Ohio, and therefore results represent an accurate performance indicator of hue- and texture-based grass identification in a real-world environment. Second, data collection times are representative of typical times people tend to mow their lawns. Both data sets maintained constant lighting conditions, where the sun was visible at all times and never obscured by cloud cover when images were recorded.

By randomly selected neighborhoods from both data sets, calculated statistics are valid for multiple times of day and are therefore not tuned to function only under specific, time-dependent lighting conditions.

[3] For further details on data collection methods and the collected data set, please see the Appendix.

## 4.2 Quantification of Descriptors

Hue and texture statistics were calculated for four square neighborhood sizes -- 11x11, 21x21, 31x31, and 41x41 pixels. After calculating the statistics for the images, the mean and standard deviation ($\sigma$) of the statistics for each type of lawn surface at each neighborhood size were computed to observe groupings in the measurements. For illuminated grass, the mean hue value, $\bar{h}$, for a 31x31 neighborhood was determined to be

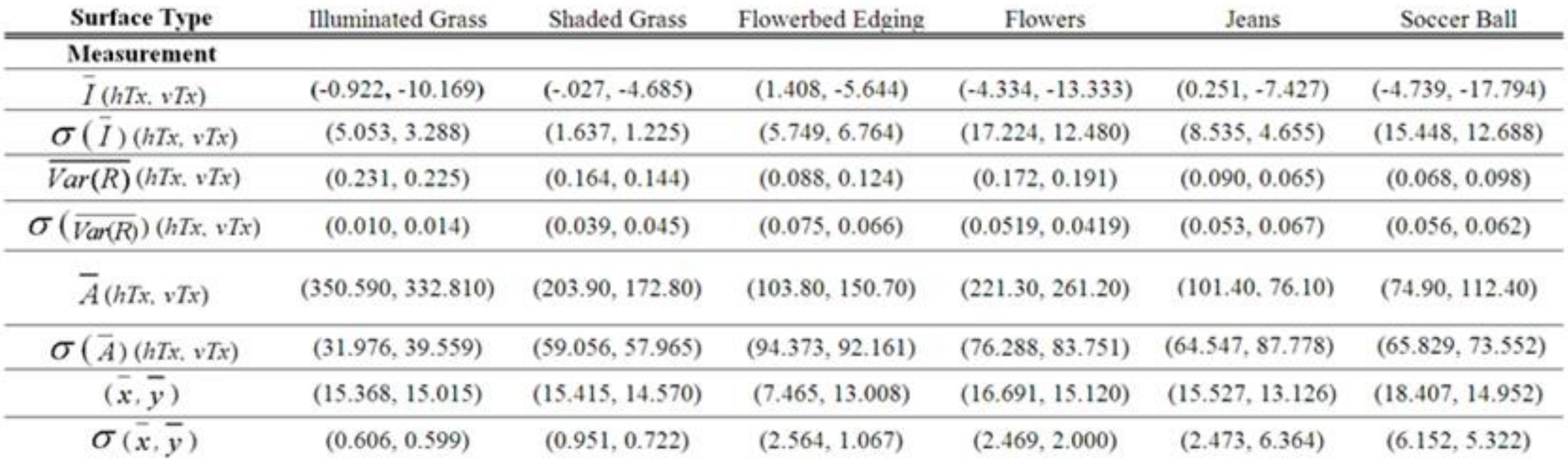
$$\bar{h} = 180.04 \tag{16}$$

with a standard deviation, $\sigma(\bar{h})$, of

$$\sigma(\bar{h}) = 30.08\,. \tag{17}$$

Table 1 displays the calculated texture statistics for sample 31x31 pixel neighborhoods [TABLE 1]. Statistics were calculated from the following number of samples for each texture type: Illuminated grass – 40 samples, shaded grass, flowers, and artificial obstacles – 10 samples each.

To visually display clustering, similar texture measures for each type of lawn surface were plotted in the same graph [FIGURES 15-30]. Graphs were generated for all statistical measurements across all four neighborhood sizes. The following data was plotted for each neighborhood size: Mean neighborhood intensity of grayscale horizontal (*hTx*) vs. vertical (*vTx*) texture images for corresponding neighborhoods ($\bar{I}$), variance of binary horizontal vs. vertical texture images for corresponding neighborhoods ($\overline{Var(R)}$), mean edge response area of binary horizontal vs. vertical texture images for corresponding neighborhoods ($\overline{A}$), and local horizontal and vertical centroid location for corresponding neighborhoods of directionally ambiguous binary texture images ($\overline{x}, \overline{y}$).



| Surface Type | Illuminated Grass | Shaded Grass | Flowerbed Edging | Flowers | Jeans | Soccer Ball |
|---|---|---|---|---|---|---|
| **Measurement** | | | | | | |
| $\bar{I}$ $(hTx, vTx)$ | (-0.922, -10.169) | (-.027, -4.685) | (1.408, -5.644) | (-4.334, -13.333) | (0.251, -7.427) | (-4.739, -17.794) |
| $\sigma(\bar{I})$ $(hTx, vTx)$ | (5.053, 3.288) | (1.637, 1.225) | (5.749, 6.764) | (17.224, 12.480) | (8.535, 4.655) | (15.448, 12.688) |
| $\overline{Var(R)}$ $(hTx, vTx)$ | (0.231, 0.225) | (0.164, 0.144) | (0.088, 0.124) | (0.172, 0.191) | (0.090, 0.065) | (0.068, 0.098) |
| $\sigma(\overline{Var(R)})$ $(hTx, vTx)$ | (0.010, 0.014) | (0.039, 0.045) | (0.075, 0.066) | (0.0519, 0.0419) | (0.053, 0.067) | (0.056, 0.062) |
| $\bar{A}$ $(hTx, vTx)$ | (350.590, 332.810) | (203.90, 172.80) | (103.80, 150.70) | (221.30, 261.20) | (101.40, 76.10) | (74.90, 112.40) |
| $\sigma(\bar{A})$ $(hTx, vTx)$ | (31.976, 39.559) | (59.056, 57.965) | (94.373, 92.161) | (76.288, 83.751) | (64.547, 87.778) | (65.829, 73.552) |
| $(\bar{x}, \bar{y})$ | (15.368, 15.015) | (15.415, 14.570) | (7.465, 13.008) | (16.691, 15.120) | (15.527, 13.126) | (18.407, 14.952) |
| $\sigma(\bar{x}, \bar{y})$ | (0.606, 0.599) | (0.951, 0.722) | (2.564, 1.067) | (2.469, 2.000) | (2.473, 6.364) | (6.152, 5.322) |

TABLE 1: Neighborhood statistics for neighborhood size of 31 (n=40: illuminated grass, n=10 others)

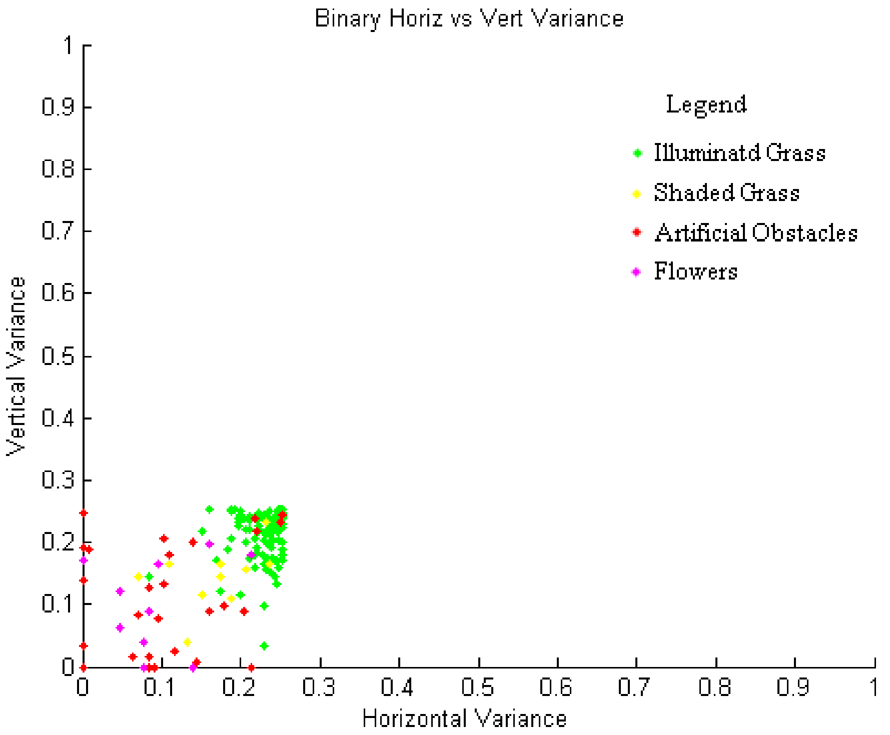


FIGURE 15: Binary horizontal vs. vertical neighborhood variance for samples at a neighborhood size of 11.

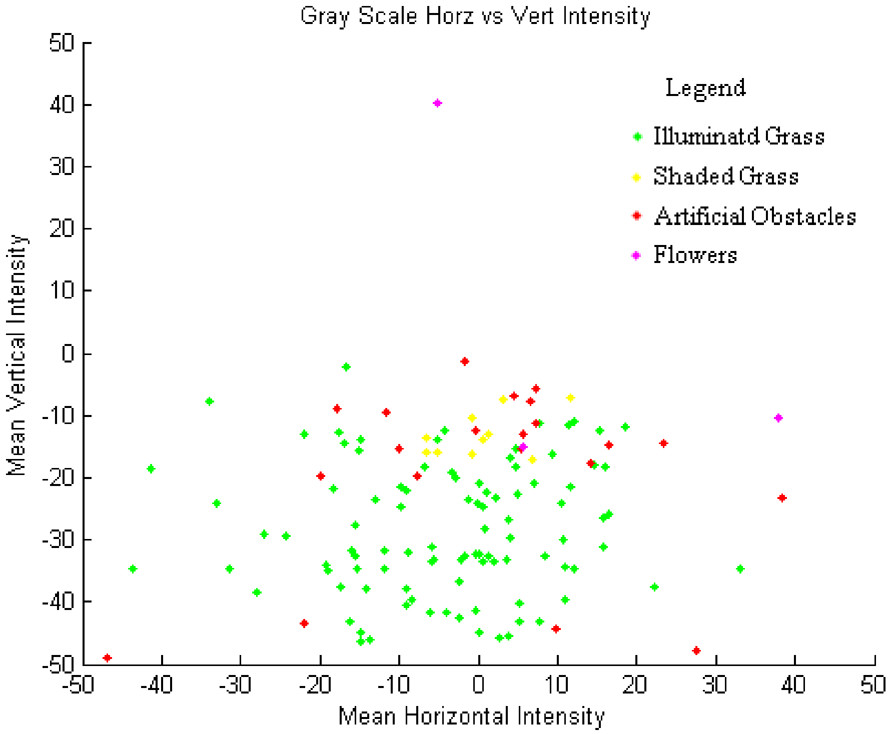


FIGURE 16: Grayscale horizontal vs. vertical neighborhood intensity for samples at a neighborhood size of 11.

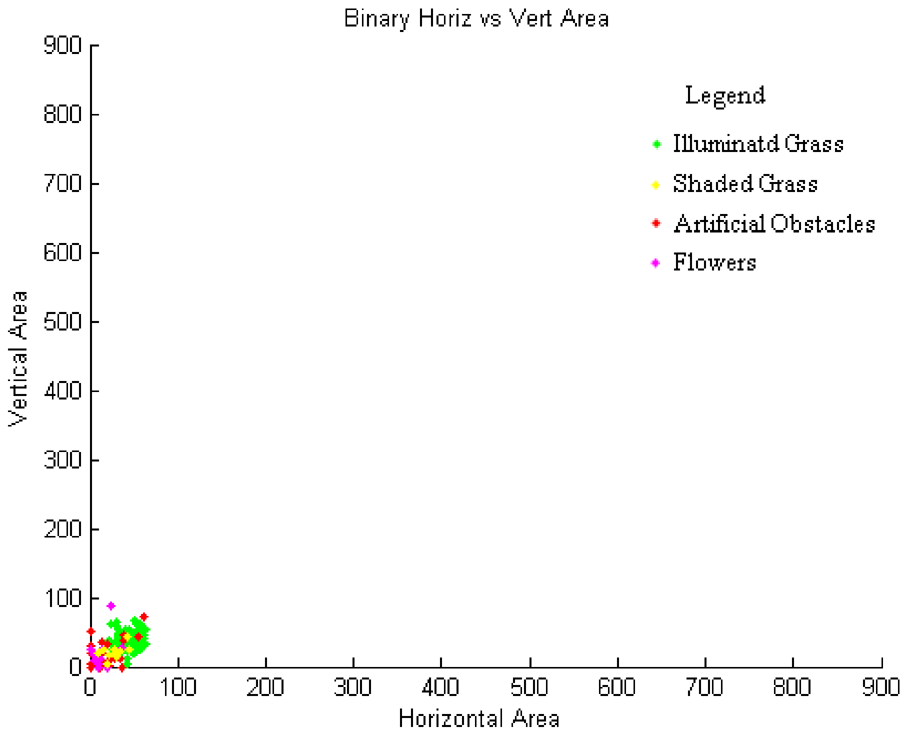


FIGURE 17: Binary horizontal vs. vertical neighborhood area for samples at a neighborhood size of 11.

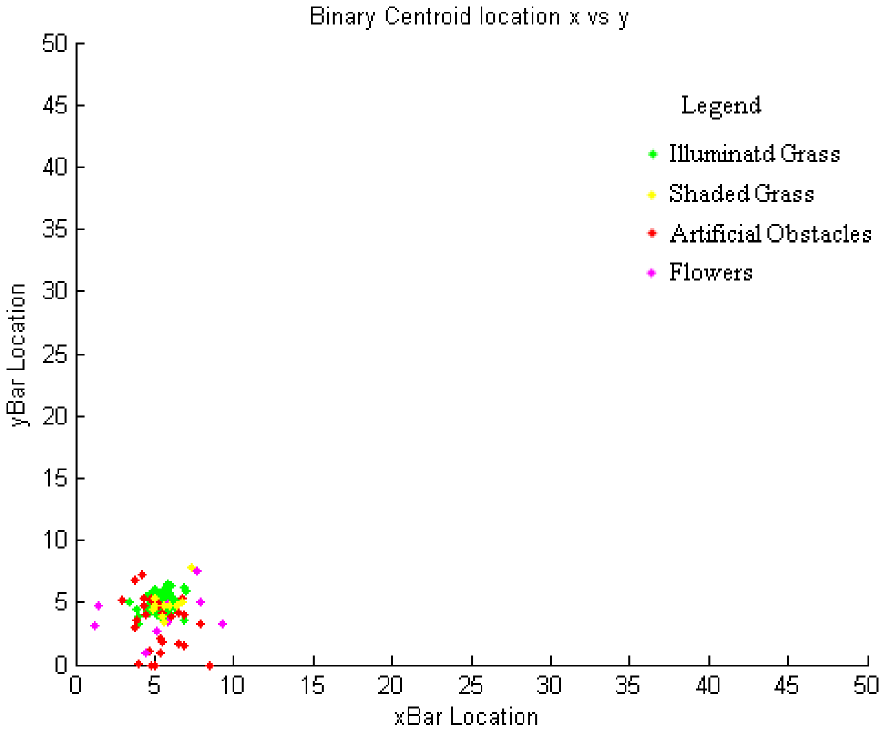


FIGURE 18: Binary x vs. y neighborhood centroid location for samples at a neighborhood size of 11.

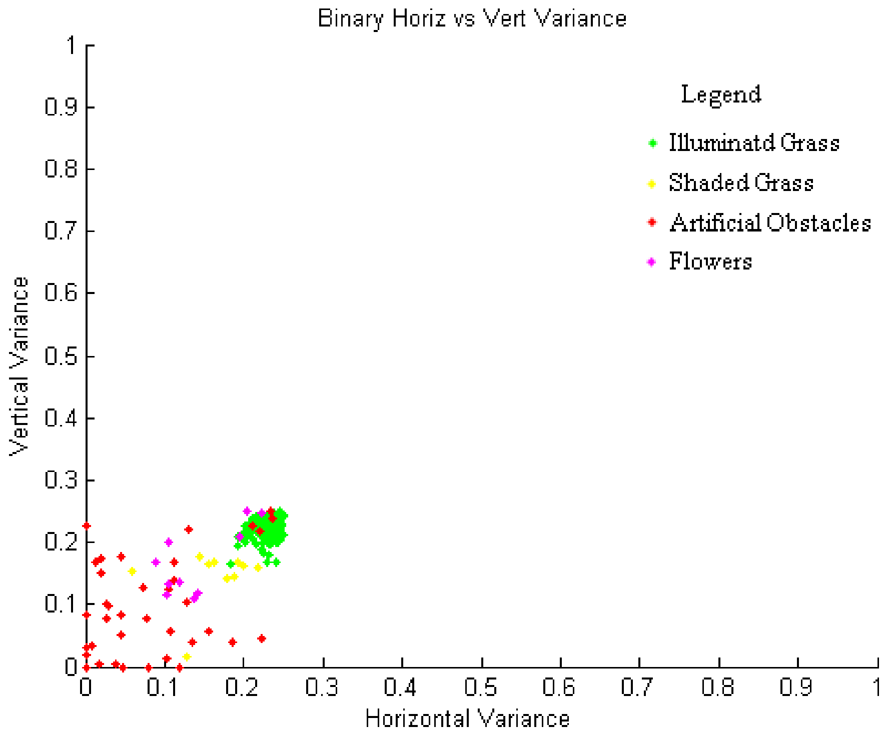


FIGURE 19: Binary horizontal vs. vertical neighborhood variance for samples at a neighborhood size of 21.

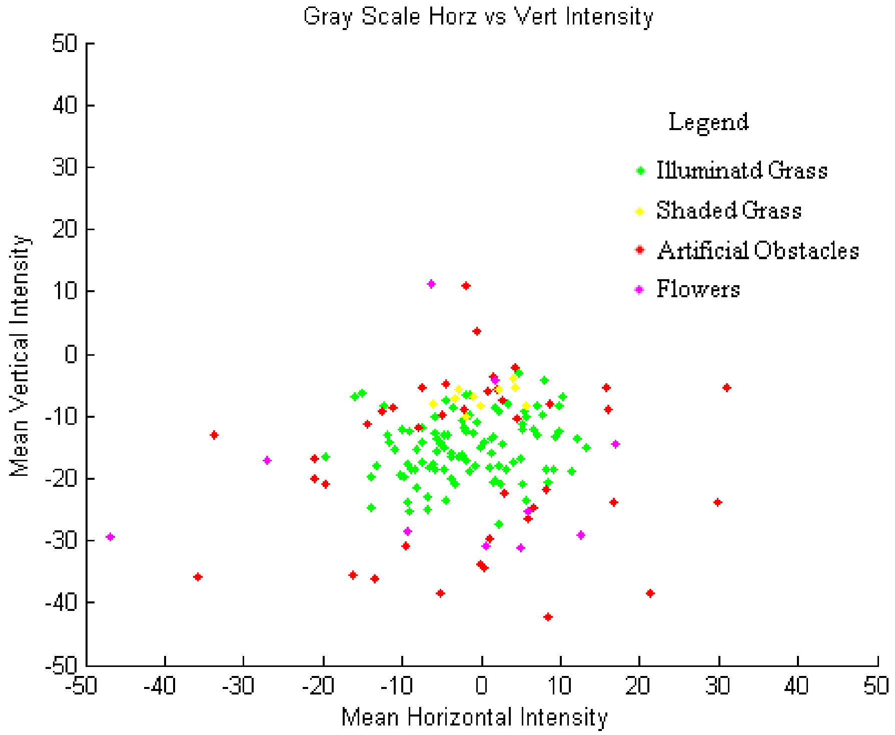


FIGURE 20: Grayscale horizontal vs. vertical neighborhood intensity for samples at a neighborhood size of 21.

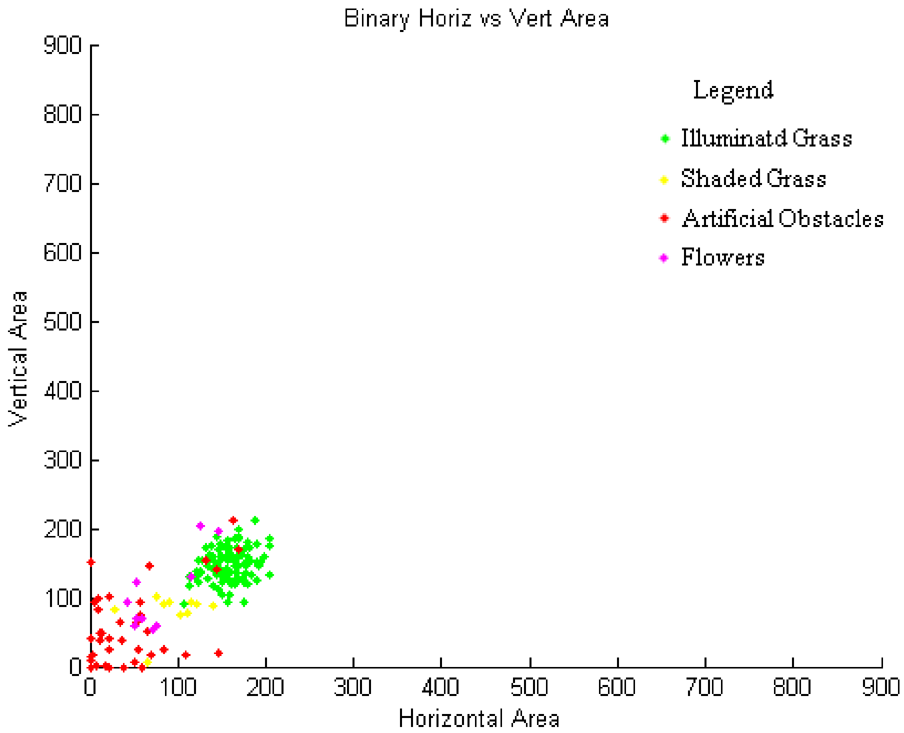


FIGURE 21: Binary horizontal vs. vertical neighborhood area for samples at a neighborhood size of 21.

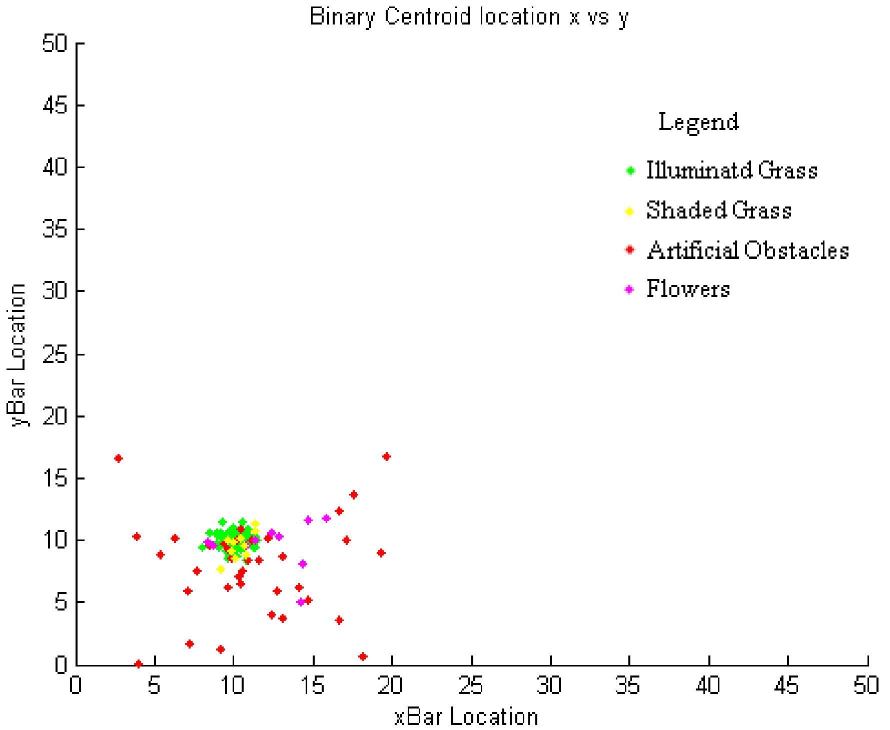


FIGURE 22: Binary x vs. y neighborhood centroid location for samples at a neighborhood size of 21.

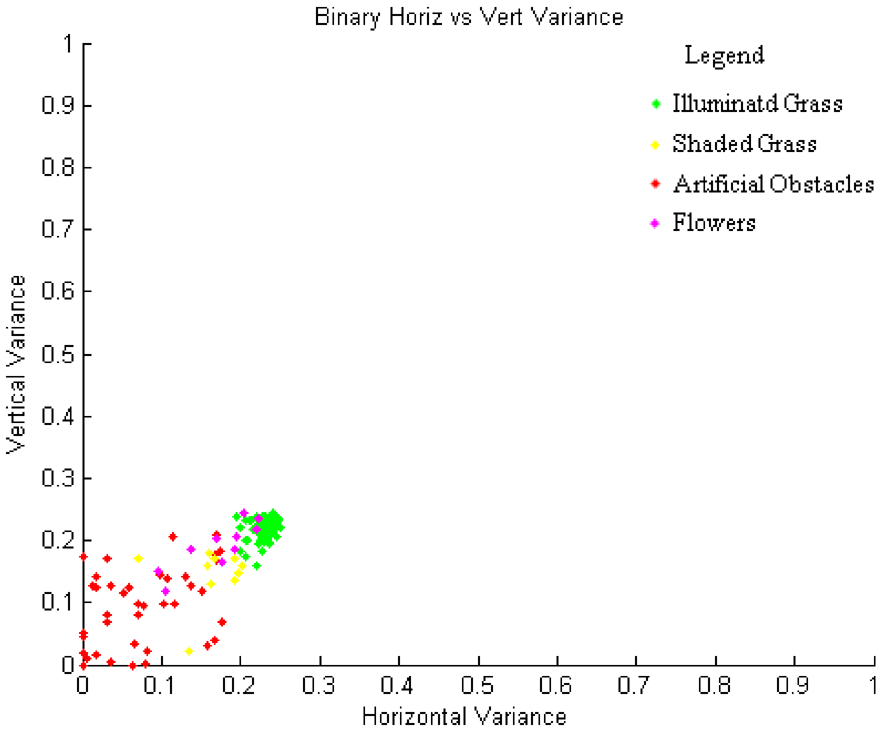


FIGURE 23: Binary horizontal vs. vertical neighborhood variance for samples at a neighborhood size of 31.

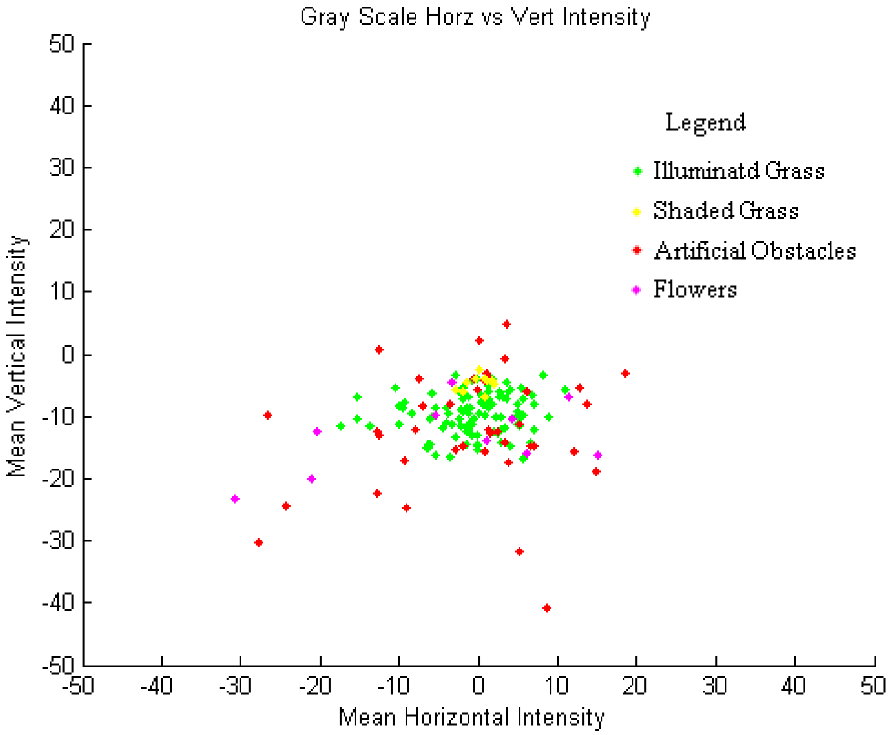


FIGURE 24: Grayscale horizontal vs. vertical neighborhood intensity for samples at a neighborhood size of 31.

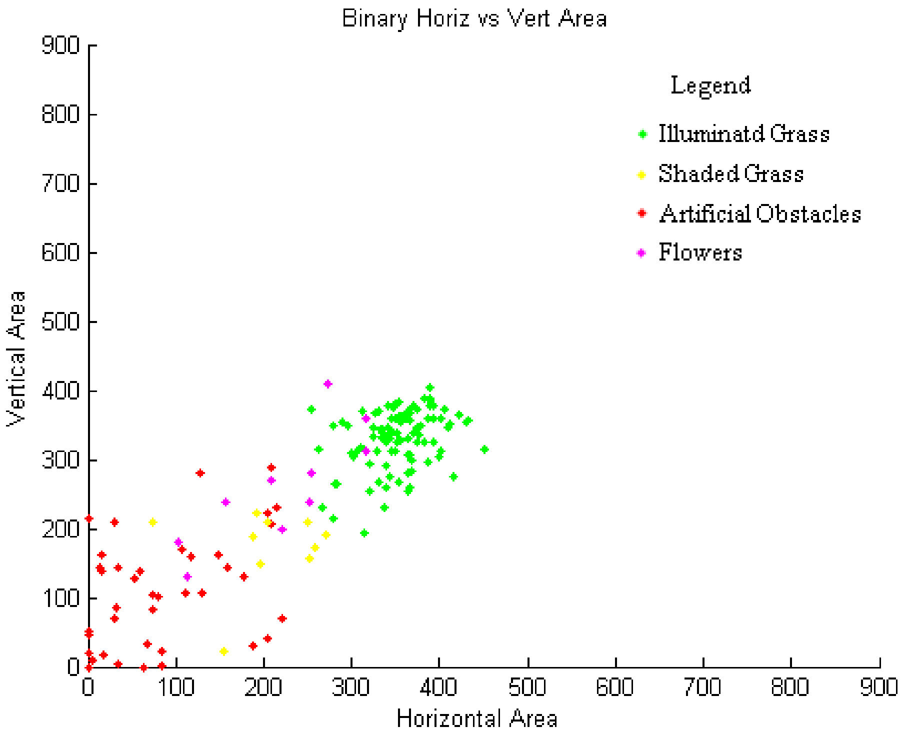


FIGURE 25: Binary horizontal vs. vertical neighborhood area for samples at a neighborhood size of 31.

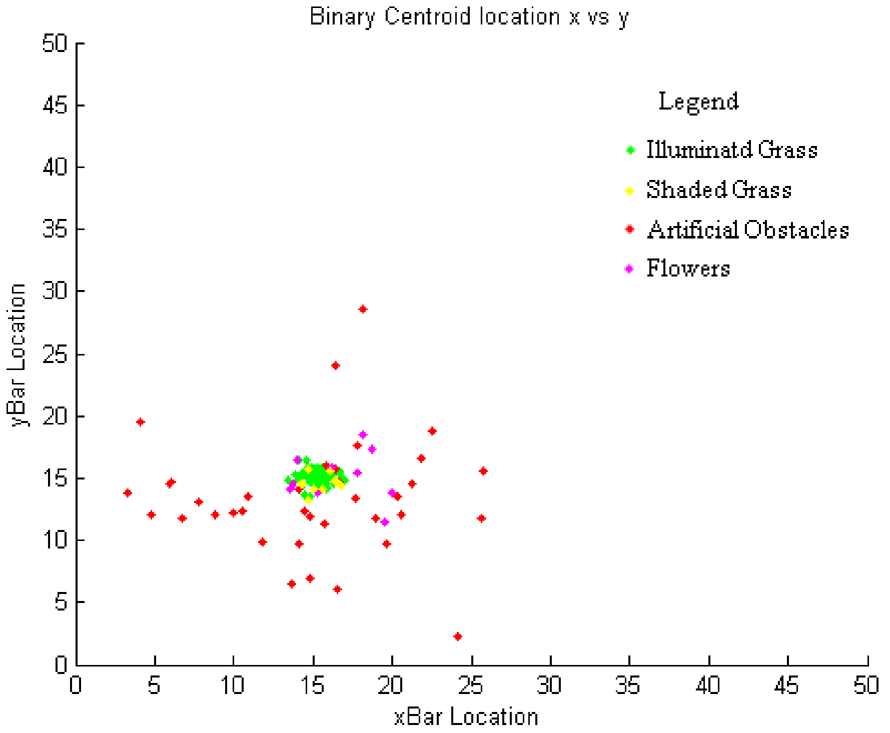


FIGURE 26: Binary x vs. y neighborhood centroid location for samples at a neighborhood size of 31.

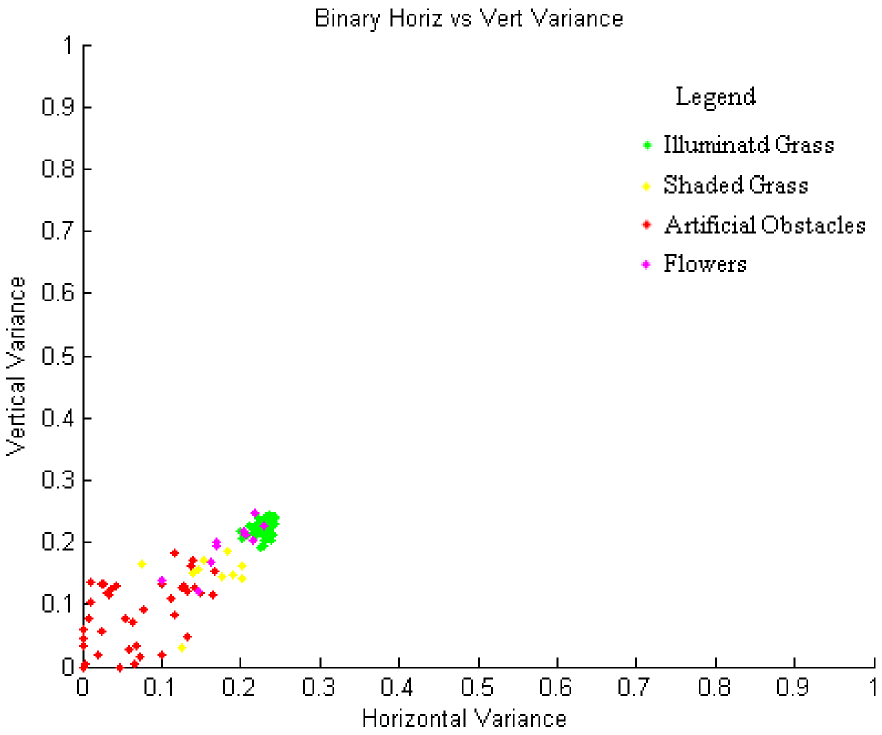


FIGURE 27: Binary horizontal vs. vertical neighborhood variance for samples at a neighborhood size of 41.

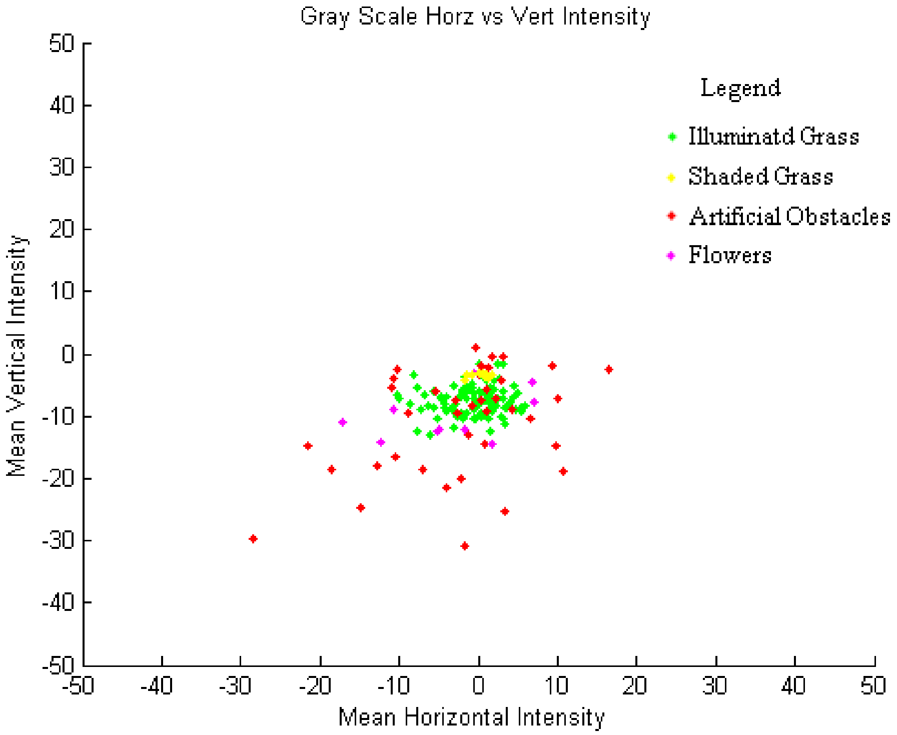


FIGURE 28: Grayscale horizontal vs. vertical neighborhood intensity for samples at a neighborhood size of 41.

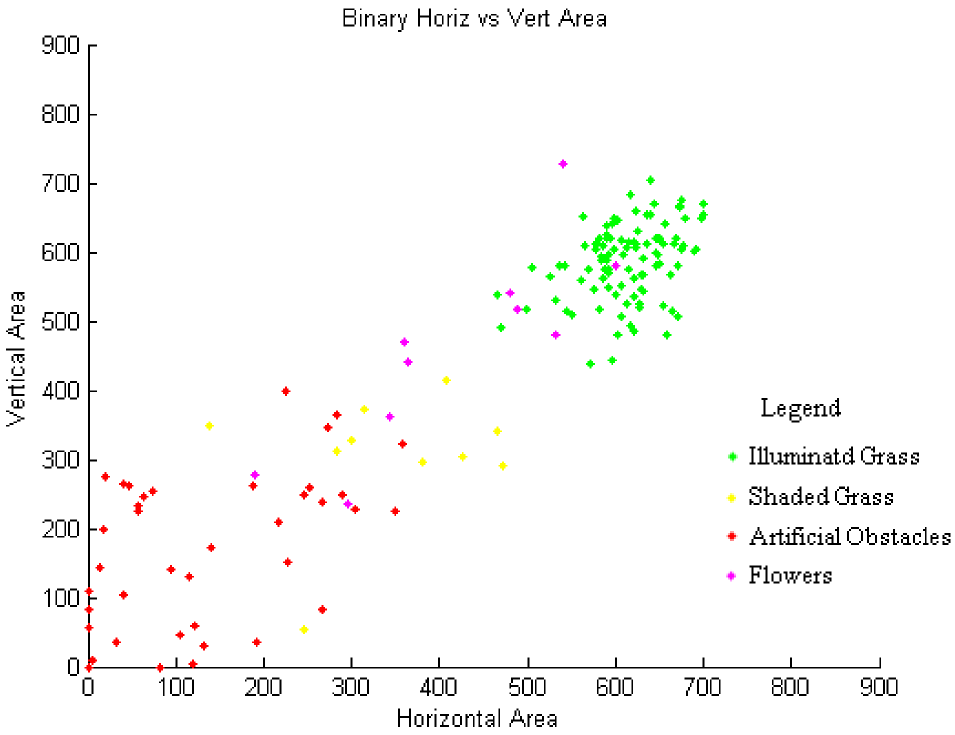


FIGURE 29: Binary horizontal vs. vertical neighborhood area for samples at a neighborhood size of 41.

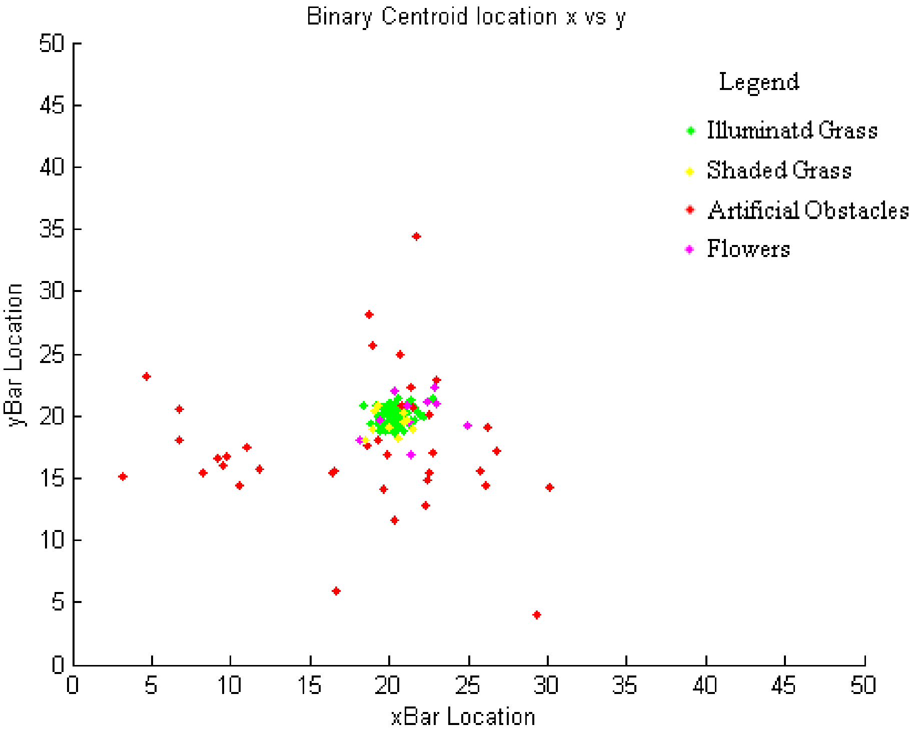


FIGURE 30: Binary x vs. y neighborhood centroid location for samples at a neighborhood size of 41.

Since no attempt was made to distinguish between different types of obstacles, all non-grass data points with the exception of flowers were plotted with the same color (red) for visualization purposes. The visual texture of fence, flowerbed edging, blue jeans, and soccer ball is referred to as “artificial texture,” as these objects are manmade. Similarly, these obstacles are described as “artificial.” In all plots, surface types were identified by the following color scheme: green – illuminated grass, yellow – shaded grass, magenta – flowers, red – artificial obstacles.

## 4.3 Discussion of Individual Texture Measurements

For the binary statistics (binary horizontal and vertical variance, area, and centroid location) distinct, compact clustering of illuminated grass was observed for all neighborhood sizes [FIGURES 15, 17-19, 21-23, 25-27, 29-30]. The mean values of the clusters for oriented texture measurements are close to the diagonal of the plots, indicating no strong directionality for either the horizontal or vertical texture images. This indicates that both horizontal and vertical texture images identify illuminated grass equally well, and that the edge filter orientation is not a determining factor in being able to identify grass based on observed statistics.

For all texture measures, the distribution of illuminated grass data points for corresponding texture measurements became more compact as the neighborhood size increased [FIGURES 15-30]. This is because computations over a larger neighborhood are less sensitive to variations in individual pixels, thereby decreasing the overall standard deviation of the neighborhood computations.

Binary horizontal and vertical measurements for oriented binary texture images of grass exhibited distinct clustering behavior as compared to artificial obstacles [FIGURES 15, 19, 23, 27]. The color uniformity of artificial obstacles (i.e white fence, black obstacle edging, blue jeans, black and white soccer ball) resulted in data points near the origin. Images of an artificial obstacle border exhibit strong unidirectional texture response. This is observed in data points located on or near one of the axes [FIGURES 15, 19, 23, 27]. Variance from artificial obstacles was distributed uniformly between the origin and approximately (0.20, 0.20) for all resolutions [FIGURES 15, 19, 23, 27]. Conversely, illuminated grass had significantly different, tightly clustered variation for both horizontal and vertical binary texture images [FIGURES 15, 19, 23, 27]. The observed variance of illuminated grass was the same for horizontal and vertical binary texture and did not change as neighborhood size increased, being centered at approximately (0.25, 0.25) [FIGURES 15, 19, 23, 27]. Noticeable separation between illuminated grass and artificial obstacles again occurred at neighborhood sizes of 31x31 or greater [FIGURES 15, 19, 23, 27].

The mean binary area exhibited separation for all neighborhood sizes [FIGURES 17, 21, 25, 29]. As the neighborhood size increased, the edge response area of illuminated grass increased, whereas the areas of artificial obstacles remained constant due to low edge response. Distinguishable separation of artificial obstacles and illuminated grass areas was again observed for neighborhood sizes greater than or equal to 31x31.

For all neighborhood sizes, tight clustering of neighborhood centroid location was observed for illuminated grass, whereas artificial obstacles did not exhibit tight clustering [FIGURE 18, 22, 26, 30].

All statistics showed overlap between the flower obstacle and shaded grass [FIGURES 15-30]. Flower textures were generally distinct from artificial obstacle textures, being either located in a different region in the graph or exhibiting a tighter clustering than the artificial textures. Both of these behaviors are attributed to the same phenomenon. While illuminated grass was the most "rough" texture in observed images, because it contained many edges per neighborhood, artificial obstacles were the "smoothest", having little color variation, which caused a low edge response. While flowers were less "rough" than illuminated grass, they were nearly as "smooth" as shaded grass in grayscale and binary textures. Performance of binary texture distinction and clustering of shaded grass did not change significantly as the edge threshold was lowered [FIGURE 31, left]. Compared to Figure 23 [reproduced in FIGURE 31, right], which plots the binary x vs. y neighborhood variance of samples at a neighborhood size of 31 and a binary edge threshold value of 21, the clustering did not significantly change as the edge threshold value was lowered. This behavior was also observed in other plots of other binary texture measurements.

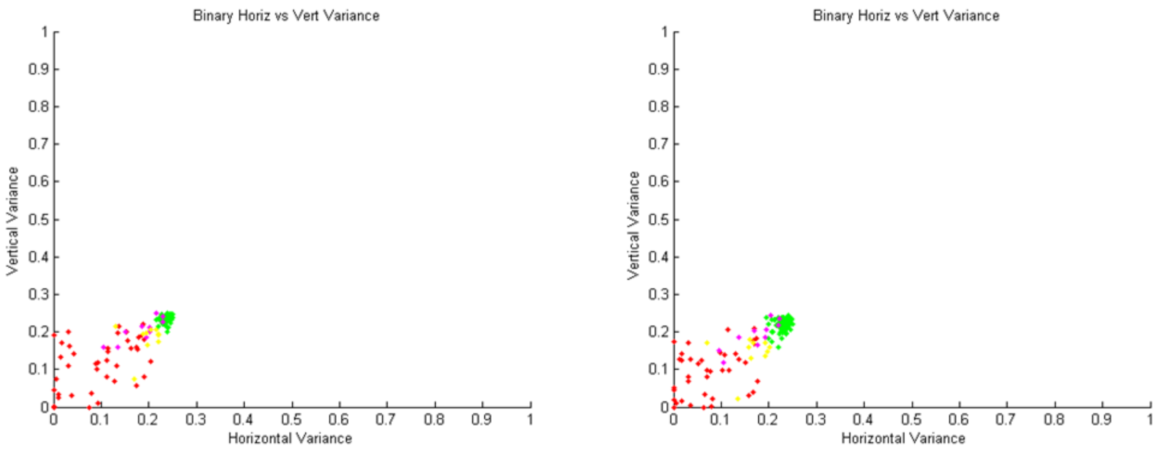


FIGURE 31: Adjusted edge thresholds. Binary x vs. y neighborhood variance for samples at a neighborhood size of 31 and a binary edge threshold value of 10 (left) and a binary edge threshold value of 21 (right).

## 4.4 Grass Identification Based on Observed Image Statistics

Though observed grass hue and all texture statistics exhibit clustering behavior, it is important to quantify their ability to correctly identify grass and differentiate between grass and obstacles. Grass identification using hue and texture statistics was tested on 40 randomly selected samples (described in 4.1) of 31x31 neighborhoods containing: illuminated grass, shaded grass, artificial obstacles, and flowers. These results are tabulated in Table 3, which lists the number of correctly identified regions [TABLE 2].

| **Measurement** | Hue | $\overline{Var(R)}\,hTx$ | $\overline{Var(R)}\,vTx$ | $\overline{A}\,hTx$ | $\overline{A}\,vTx$ | $(\bar{x},\bar{y})$ | $\overline{I}\,hTx$ | $\overline{I}\,vTx$ |
|---|---|---|---|---|---|---|---|---|
| **Illuminated Grass** | 40 | 40 | 40 | 38 | 38 | 40 | 38 | 14 |
| **Shaded Grass** | 21 | 36 | 36 | 36 | 36 | 39 | 40 | 8 |
| **Artificial Obstacles** | 32 | 37 | 31 | 35 | 35 | 28 | 10 | 33 |
| **Flowers** | 16 | 11 | 5 | 9 | 9 | 21 | 14 | 27 |

TABLE 2: Number of correctly identified 31x31 neighborhoods for each statistic (n=40 for each measurement).

Identification rates were determined as follows. Based on figures 15-30, illuminated grass was distinctly clustered from artificial obstacles for neighborhood sizes greater than 31x31. Additionally, shaded grass clusters were always in the vicinity of illuminated grass clusters for corresponding texture measurements [FIGURES 15-30]. If a neighborhood's statistic fell within three standard deviations of the corresponding measurement's mean for illuminated grass, the neighborhood was marked as grass. Failures in identifying illuminated and shaded grass occurred when a neighborhood containing only these surfaces fell outside of the illuminated grass range. Similarly, failures in identifying artificial obstacles and flowers occurred when a neighborhood containing only these obstacles fell within the illuminated grass range (false positives). Correct identification of illuminated and shaded grass occurred when a neighborhood containing only these surfaces fell within the range of illuminated grass. Correct identification of artificial obstacles and flowers occurred when a neighborhood containing only these surfaces fell outside of the range of illuminated grass.

Illuminated grass was correctly identified 40/40 times (100% accuracy) using only hue. Hue correctly identified 21/40 samples (52% accuracy) of shaded grass, 32/40 (80%) of artificial obstacles, and (16/40) 40% of flowers correctly. Illuminated grass was correctly identified with at least 38/40 (95% accuracy) for all texture measurements except mean vertical grayscale intensity. Shaded grass was identified with at least 36/40 (90% accuracy) for all texture measurements except the mean vertical grayscale intensity. Binary horizontal variance and binary horizontal and vertical area identified obstacles correctly with at least 31/40 (77.5% accuracy). Flowers were poorly identified. This may be due to texture similarity between the grass and flowers. It is important to note that this performance may not be indicative of recognition ability for all flowers, since only images of one flower type (*Chrysanthemum*) was used for analysis [FIGURE 32].

FIGURE 32: A chrysanthemum.

## 4.5 Grass Identification Based on Combined Statistics

Based on the results in Section 4.4, a weighting coefficient for each statistic was calculated using (14) and is tabulated in Table 3 [TABLE 3]. Since no statistics were able to differentiate between flowers and obstacles with at least a 30/40 (75%) success rate, the flower identification results were not factored into the coefficient calculations. This is justifiable for operation in the ION environment where all of the flowers are surrounded by plastic edging material (an obstacle), which was consistently recognized by the majority of the measurements.

| **Weight Coeff.** | Hue | $\overline{Var(R)}\,hTx$ | $\overline{Var(R)}\,vTx$ | $\bar{A}\,hTx$ | $\bar{A}\,vTx$ | $(\bar{x},\bar{y})$ | $\bar{I}\,hTx$ | $\bar{I}\,vTx$ |
|---|---|---|---|---|---|---|---|---|
| **α** | 0.7750 | 0.9416 | 0.8916 | 0.9083 | 0.9083 | 0.8916 | 0.7333 | 0.45833 |

TABLE 3: Pre-weight coefficients for each statistic.

Grass identification via these combined statistics was tested on 40 randomly selected samples of 31x31 neighborhoods containing the same objects that were used for individual texture measurements. The combined statistical measurements relied on six identifiers – hue, $\overline{Var(R)}_{hTx}$, $\overline{Var(R)}_{vTx}$, $\overline{A}_{hTx}$, $\overline{A}_{vTx}$, and $(\bar{x}, \bar{y})$. Intensity measurements were not included, due to the fact that they were not able to identify grass and obstacles correctly with an accuracy of over 30/40 (75%). The weighting coefficients were normalized for these six identifiers and are presented in Table 4 [TABLE 4].

| Normalized Coeff. | Hue | $\overline{Var(R)}_{hTx}$ | $\overline{Var(R)}_{vTx}$ | $\overline{A}_{hTx}$ | $\overline{A}_{vTx}$ | $(\bar{x}, \bar{y})$ |
|---|---|---|---|---|---|---|
| **αNorm** | 0.1458 | 0.1771 | 0.1677 | 0.1708 | 0.1708 | 0.1677 |

TABLE 4: Normalized pre-weight coefficients for six identifiers.

Correct neighborhood identification was deemed to occur if the neighborhood was correctly identified as grass or obstacle containing with a voting agreement greater than 0.5 (50%). Since each of the utilized measurement had approximately the same individual accuracy, this corresponds to the majority of the neighborhoods identifying the neighborhood as either grass containing or obstacle containing. Neighborhood identification rates were also calculated for a voting agreement greater than 0.75 (75%). The number of correctly identified neighborhoods based on this criterion of the combined measurement for the four neighborhood types is tabulated in Table 5 [TABLE 5].

| Measurement | Combined (50% Voting Threshold) | Combined (75% Voting Threshold) |
|---|---|---|
| **Grass** | 40 | 40 |
| **Shadow** | 38 | 33 |
| **Artificial Obstacles** | 32 | 38 |
| **Flowers** | 6 | 15 |

TABLE 5: Number of correctly identified 31x31 neighborhoods for combined measurement (n=40 for each measurement).

For 0.5 (50%) voting agreement or greater, the ability to correctly identify shaded grass increases compared to nearly all statistics to 38/40 (95%) when combining multiple texture measurements. Illuminated grass is still recognized with 40/40 (100%) accuracy. Artificial obstacle recognition performance also increases compared to stand-alone hue and binary vertical variance measurements. For 0.75 (75%) voting agreement or greater artificial obstacle recognition performance improves significantly to 38/40 (95%) and illuminated grass is still recognized with 40/40 (100%) accuracy. Shaded grass identification drops to an accuracy of 33/40 (82.5%). This drop occurs, because for a 0.75 threshold to be reached for the neighborhood during voting, more individual measures have to agree that the neighborhood contains grass.

Flowers remain poorly identified for both voting thresholds. This is due to poor recognition of flowers using individual texture measurements. Again, this performance may not be indicative compared to other flowers, since only one type was analyzed.

## 4.6 Comparison to Similar Identification Algorithms

The results of the grass identification technique presented in this thesis cannot be directly compared to other related methods, since this method presents a novel approach and application to identifying grass in images and performing image segmentation.

Zafarifar et al. developed a method which combines color and texture measurements to determine grass locations in images of football fields [29]. This is used for grass-color correction and digital image overlays during television broadcasts. While this application does not examine grass at as fine a texture as the method presented in this thesis and the grass and lighting conditions are more strictly controlled since measurements are calculated for and applied in a sports venue setting, Zafarifar et al.'s method is appropriate for comparison purposes since it combines texture and color measurements to determine grass locations in images. It is important to note that whereas Zafarifar identifies grass based on its probability, the method presented in this thesis utilizes a voting scheme. Zafarifar's method is able to identify 91% of grass in images correctly.

Another appropriate comparison can be made between the presented method and Watchareeruetai's method, which also combines color and texture measurements to determine weed locations in images for targeted pesticide spraying [26]. Watchareeruetai et al. also utilize neighborhood statistics of binary texture images and combine this with hue measurements to determine weed locations. However, their method differs in several ways. First, they attempt to create pixel-level-accurate representations of weeds by utilizing overlapping neighborhoods. Additionally, their algorithm does not run on their robot in real-time and is computed after all of the data has been collected. Their method is able to identify 91.1% of weed locations in images correctly.

The method described in this thesis combines the real-time identification requirement met by Zafarifar et al. with hue and texture combination techniques similar to Watchareeruetai et al. Using hardware similar to that on the 2009 CWRU Cutter (Intel Core 2 4400 @ 2GhZ with 2GB of RAM), individual texture measurements were computed at a real-time 15KhZ speed for 21x21 pixel neighborhoods.[4] For at least a 0.5 voting agreement, identification through hue and texture combination yields a 100% accurate identification rate for illuminated grass, a 95% accurate identification rate for shaded grass, and an 80% accurate identification rate for artificial obstacles.

[4] The target operating rate for algorithms onCWRU Cutter is at least 10Hz to be considered "real-time." Utilizing six 320x240 images, each of which contains approximately 165 21x21 neighborhoods, all neighborhoods are processed at a rate of approximately 16Hz. Furthermore, this is a conservative estimate, since this is not optimized, and does not re-use common elements for multiple texture measurements, such as $M_{00}$.

## 4.7 Qualitative Analysis of Processing Results on Full Images

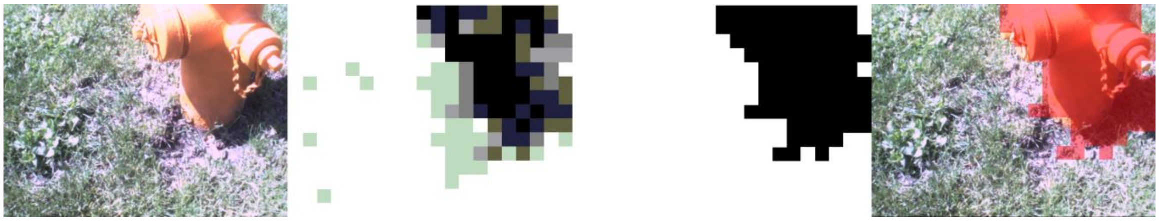


FIGURE 33: Full image processing with a fire hydrant obstacle. (Left to right) Input image, pre-thresholded voting combination, generated binary freespace, overlay of input image and freespace.

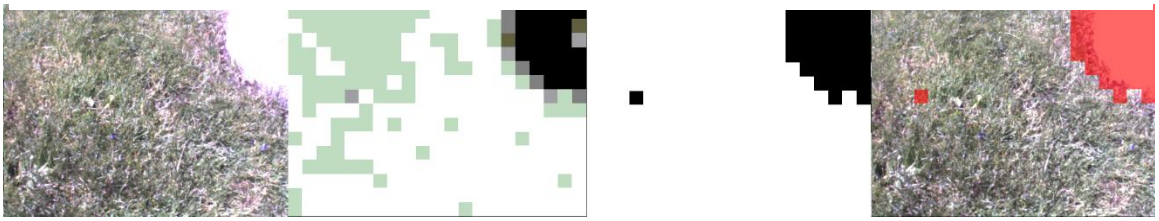


FIGURE 34: Full image processing with a concrete block obstacle. (Left to right) Input image, pre-thresholded voting combination, generated binary freespace, overlay of input image and freespace.

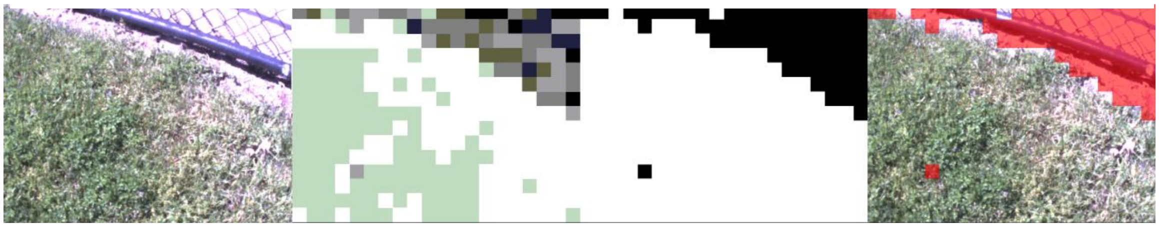


FIGURE 35: Full image processing with a fence obstacle. (Left to right) Input image, pre-thresholded voting combination, generated binary freespace, overlay of input image and freespace.

Figures 33-35 illustrate the result of applying the image processing technique presented in this thesis to full images [FIGURES 33-35]. These pictures were taken on a different day, at a different location than the images used to perform a quantitative analysis of neighborhood based identification.

In each figure, the image on the right illustrates obstacle locations predicted by the algorithm that are overlaid with the original input image to show correspondence between the predicted and real-world obstacle locations. Each method is able to identify obstacle locations with a high degree of accuracy. Though non-connected false positives exist, these could be filtered out via the methods proposed in Section 5.2.

It is important to note that the obstacles present in Figures 33-35 do not correspond to obstacle types analyzed during training. This demonstrates that being able to consistently identify grass is enough to determine obstacle locations in input images.

# 5. Conclusions

## 5.1 Summary

This thesis proposed a method to identify grass containing regions in incoming images based on computationally inexpensive measurements -- perceived hue and first- and second-order texture statistics. This information can be used by autonomous lawnmowers, such as CWRU Cutter for real-time, vision-based navigation and obstacle avoidance in lawn navigation. Based upon six measurements, images were then segmented into neighborhoods and corresponding neighborhoods were recombined to create an improved representation of grass locations. A weighted voting scheme was used to combine recognition results from hue and texture statistics. For a 50% voting agreement threshold, 100% of neighborhoods[5] containing illuminated grass, 95% of neighborhoods containing shaded grass, and 75% of neighborhoods containing artificial obstacles, such as white plastic fence and plastic flowerbed edging were correctly identified. For a 75% voting threshold, 100% of neighborhoods containing illuminated grass, 82.5% of neighborhoods containing shaded grass, and 95% of neighborhoods containing artificial obstacles were correctly identified. Due to their color and texture similarity compared to grass, flowers were poorly identified at both probabilities, being identified with less than 40% accuracy.

[5] n=40: illuminated grass, n=40: shaded grass, n=40: artificial obstacles

## 5.2 Future Work

While this thesis demonstrated that it is possible to differentiate between grass and various other obstacles, the currently implemented image segmentation is relatively simple. Several approaches could be taken to advance the segmentation method, which could improve overall recognition performance, especially between grass and other natural textures.

First, individual binary freespace representations could be post processed before combination in order to eliminate clear false positives in the images. In captured camera images, if a neighborhood is identified as obstacle containing, but all of its surrounding neighbors are identified as grass, it is likely that the obstacle neighborhood was falsely identified, since CWRU Cutter would not encounter such small obstacles during mowing. By comparing a neighborhood to its neighbors, additional information is gained about what is observed in the image and recognition performance could be improved.

Additionally, overlapping neighborhood segmentation similar to Watchareeruetai et al. could be applied instead of the non-overlapping neighborhoods that are currently used. By using overlapping neighborhoods, the resolution of the freespace images increases and the size of false positive neighborhoods could be reduced. This again could increase recognition performance, since a larger percentage of grass in the image may be correctly identified. It is important to note, however, that by using overlapping neighborhoods, the number of computations required to process an image increases, since some pixels in the image are now examined more than once. This could be detrimental for real-time identification of grass on an autonomous lawnmower, which may not be fast enough to process these images.

The statistical quantification and segmentation methods could be changed. By modeling color and texture statistics as a mean and standard deviation pair, it is assumed that these statistics follow a Gaussian distribution. This may not be true. Instead, it is likely that statistical groupings are defined by a more complex shape, whose boundaries could be better defined by a polynomial fit. The fact that texture statistics are not simple Gaussian distributions has already been suggested by the overlap of statistical groupings for texture identifiers of various objects [FIGURES 15-30]. These groupings could be identified by using the Mahalanobis distance, which is a pattern classification method that utilizes the means of and correlations between various measurements to determine the clustering of various measurements [30]. By identifying groupings and modifying the thresholds used to identify grass in an image to reflect this, recognition performance could be drastically improved.

Along these lines, groupings across multiple statistics in higher-dimensional space could be analyzed to determine if clustering exists for combinations of multiple statistics. This could again result in a clearer grouping, whose shape could be classified utilizing techniques such as the Mahalanobis distance to determine improved statistical thresholding ranges.

To further expand this research, it is necessary to analyze algorithm performance on other test data in the database, which was not restricted to one type of grass and one lighting condition. Based on the behavior observed in datasets that were taken at hour increments throughout the day and over multiple days and common grass types, it may be possible to create an algorithm which adaptively adjusts threshold values for color and texture based on the date, time of day, and grass type.  By quantifying how changing these variables affects threshold values, it is possible to create a grass recognition method which could be utilized on an autonomous lawnmower in an applied environment.

# Appendix

## A.1 Data Collection

A set of 1800, 640x480 RGB images were collected at MTD Products, Inc.'s test facility, located in Medina, Ohio, during four days within a two month period [FIGURE 36]. Images were taken relative to the test day's solar noon time and spanned from dawn (6.5 hours before solar noon for the first test day) to dusk (6.5 hours after solar noon for the first test day), to catalog many of the lighting conditions an autonomous lawnmower could encounter in a typical home environment.

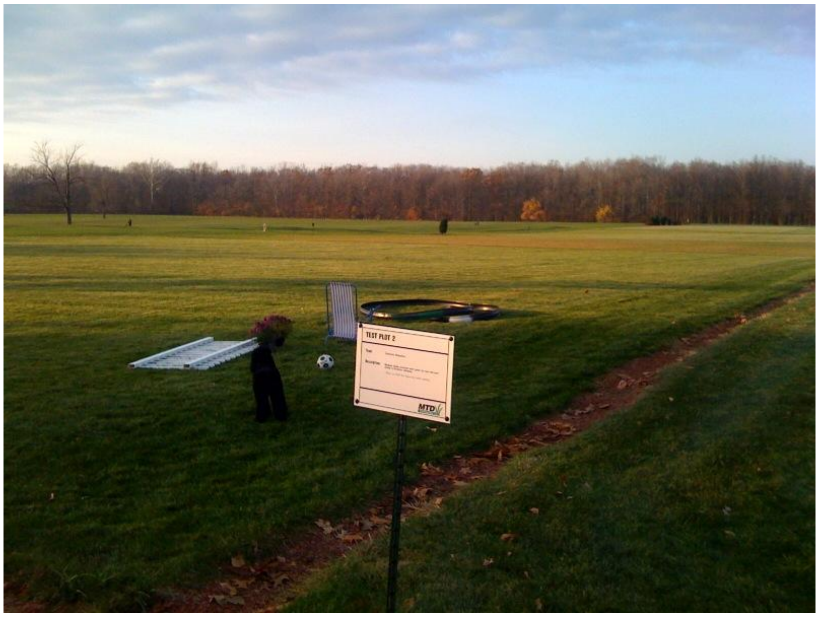


FIGURE 36: Plots of various grass types at MTD's test facility in Medina, Ohio. Image taken near dusk at the end of the second test day.

Since the day length, defined as the time when the sun is first visible on the horizon to when it disappears beneath the horizon line, decreased with each testing day, collection times for subsequent days after the first test day were adjusted to account for this fact. For example: the day length during the first test day was 13 hours, 8 minutes, and 58 seconds, with solar noon occurring at 13:27:03 for Medina's geographic location. By the second test day, the day length had decreased by approximately 49 minutes, lasting 12 hours, 20 minutes, and 9 seconds, with solar noon occurring at 13:21:58. By calculating the ratio of day lengths between the first and subsequent test days, it was possible to quantify a "time-shift correction factor," which was used to scale the subsequent start and end times for data collection.

Due to operator limits, it was impossible to capture all of the images for one data set simultaneously. Therefore, light levels and illumination conditions were assumed to be constant for a one-hour period between the start and end times during the first test day. Since the day length decreased over subsequent test days, the one-hour range was scaled for subsequent days by the time-shift correction factor.

All data was acquired using a camera setup which duplicated CWRU Cutter's geometry and hardware. The camera was elevated 0.75 meters at an angle of –45º to the horizontal and was fixed on a collapsible dolly cart [FIGURE 37]. The camera was an ImagingSource DFK21AF04 FireWire camera with a Kowa LM4PBR 4.0mm F1.2 CS-Mount lens.

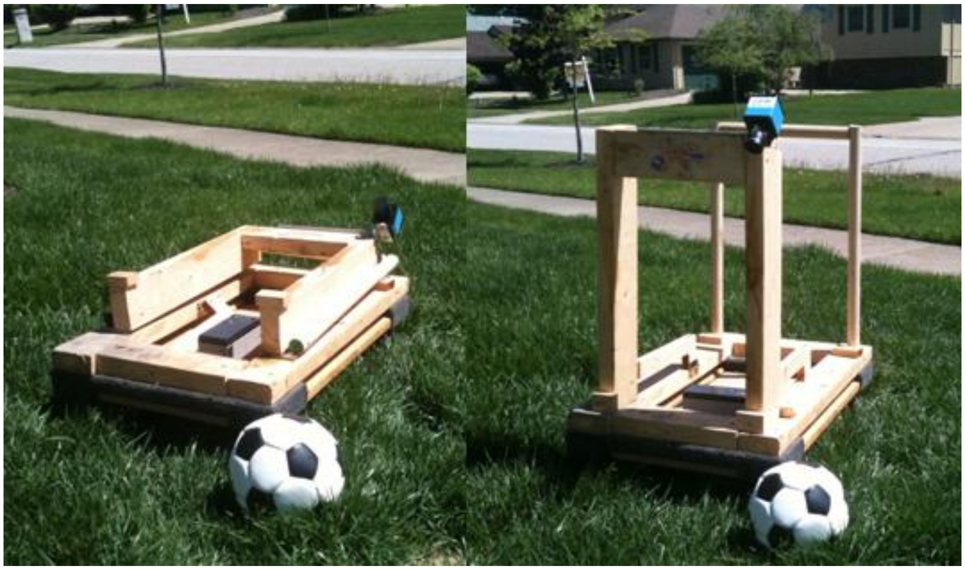


FIGURE 37: The CWRU Cutter Collapsible Chase Vehicle (CCCCV).

suburban lawns and included the following obstacles: none, shadows, flowers, flowers in flowerbed edging, blue jeans, white plastic fence, a lawnchair, a soccer ball, and a stuffed dog. The flowers, flowerbed edging, white plastic fence, and stuffed dog matched the obstacle specifications outlined by the ION competition rulebook [8]. Ten images of each obstacle type were taken at normal and parallel camera lens orientations relative to the sun for a total of 20 images of each obstacle type for each of their recorded timeframes. To ensure that edge detector response was not biased by the orientation of obstacles in captured images, the obstacle orientation was randomly changed in successive images by moving the cart around the obstacle.

## A.2 Collected Dataset Information

Test Day 1

| | |
|---|---|
| Test Date: | 8/31/2009 |
| Length of Day (Hrs:Mins:Secs): | 13:08:58 |
| Time of Solar Noon (Time): | 13:27:03 |
| Time Shift/Correction Factor: | 1 |

Test Day 2

| | |
|---|---|
| Test Date: | 9/18/2009 |
| Length of Day (Hrs:Mins:Secs): | 12:20:09 |
| Time of Solar Noon (Time): | 13:21:58 |
| Time Shift/Correction Factor: | 0.939 |

Test Day 3

| | |
|---|---|
| Test Date: | 11/23/2009 |
| Length of Day (Hrs:Mins:Secs): | 9:39:00 |
| Time of Solar Noon (Time): | 12:13:00 |
| Time Shift/Correction Factor: | 0.73477 |

Test Day 4

| | |
|---|---|
| Test Date: | 11/4/2009 |
| Length of Day (Hrs:Mins:Secs): | 10:15:21 |
| Time of Solar Noon (Time): | 12:10:00 |
| Time Shift/Correction Factor: | 0.77946768 |

Day 1 Collection Information

| Plot # | Set # | Time Shift | Actual Start Time | Missing? | Notes |
|---|---|---|---|---|---|
| 1 | 2 | 4.5 Hr Pre-Noon | 9:00:00 | Shadow | |
| 1 | 3 | 2.5 Hr Pre-Noon | 11:07:00 | | |
| 1 | 4 | Solar Noon | 13:30:00 | | |
| 1 | 5 | 2.5 Hr Post-Noon | 16:00:00 | Stuffed Dog | |

Day 2 Collection Information

| Plot # | Set # | Time Shift | Actual Start Time | Missing? | Notes |
|---|---|---|---|---|---|
| 1 | 2 | Start 4.225 Hr Pre-Noon (Begin 4.5 Hrs Pre-Noon) | 9:00:00 |  | Makeup: Day 1 Shadow |
| 4 | 3 | Start 2.3475 Hr Pre-Noon | 11:07:00 |  |  |
| 5 | 5 | End 2.3475 Hr Post-Noon | 13:30:00 |  |  |
| 1 | 5 | End 4.3475 Hr Post-Noon | 16:00:00 |  | Makeup: Day 1 Stuffed Dog |

Day 3 Collection Information

| Plot # | Set # | Time Shift | Actual Start Time | Missing? | Notes |
|---|---|---|---|---|---|
| 1 | 3 | Start 1.836 Hrs Pre-Noon (Begin 2.5 Pre-Noon) | 10:24:00 | Shadow |  |
| 2 | 5 | End 1.836 Hrs Post-Noon (End 2.5 Post-Noon) | 13:02:00 | Shadow |  |

Day 4 Collection Information

| Plot # | Set # | Time Shift | Actual Start Time | Missing? | Notes |
|---|---|---|---|---|---|
| 1 | 6 | 3.508 Hrs Post-Noon (End 4.5 Hrs Post-Noon) | 14:30:00 |  | Makeup: Day 1 End 4.5 Hrs Post-Noon |
| 1 | 7 | 4.677 Hrs Post-Noon (End 6 Hrs Post-Noon) | 15:40 |  | Makeup: Day 1 End 6 Hrs Post-Noon |

The tables labeled “Test Day” provide information about the date when corresponding data was taken. “Test date” corresponds to the day the data was taken, “length of day” states how long the day was, “time of solar noon” indicates the time of day that solar noon occurred, and “time shift/correction factor” corresponds to how start and end times for data collection had to be scaled relative to the first day’s length of day to keep lighting conditions constant.

The tables labeled “Day X Collection Information” contains information about the data taken on the corresponding days. “Plot #” indicates which MTD plot the data was collected on, “Set #” corresponds to the folder name where the data is contained in the data base, which corresponds to the time of day that the data was taken, “time shift” tabulates the adjusted start/end time of when the data was collected and in parentheses lists which start time this corresponds to from the first day, “missing” corresponds missing images of a certain surface within that data set, and “notes” correspond to notes about the data.  For example: the note “Makeup: Day 1 Shadow” in the first row of the “Day 2 Collection Information Table” states that this recording was made in order to fill in missing shadow images from the corresponding data set taken on Day 1.

## Sources Cited

[1] R.W. Hicks II, E.L Hall. “A survey of robot lawn mowers,” *Proc. of SPIE*. San Diego, USA, vol. 4719, pp. 262-269, 2000.

[2] GNU Image. “RGB Color Cube.” Retrieved from <http://en.wikipedia.org/wiki/File:RGB_color_solid_cube.png>

[3] “LB3510 Manual.” Kyodo of America, 06 18 2008. Web. 6 May 2010. <http://www.lawnbott.com/pdf/LB3510-manual.pdf>.

[4] S. Tsugawa. “A History of Automated Highway Systems in Japan and Future Issues,” 2008 IEEE Conference on Vehicular Electronics and Safety, Columbus, OH, September 2008.

[5] “DARPA Grand Challenge Commemorative Program.” Defense Advanced Research Projects Agency, Retrieved from <http://www.darpa.mil/grandchallenge04/program.pdf>, 2006.

[6] “The Purpose of IGVC.” Retrieved from <http://www.igvc.org/objective.htm>, September 20 2006.

[7] W. Newman. "Roberto (IGVC Design Report)." Retrieved from <http://www.igvc.org/design/reports/dr148.pdf>, 2006.

[8] “The Sixth Annual Robotic Lawnmower Competition Rulebook.” Retrieved from < http://www.ion.org/satdiv/alc/rules2009.pdf>, 2009.

[9] K. A. Daltorio, A. D. Rolin, J. A. Beno, B. E. Hughes, A. Schepelmann, J. Green, M. S. Branicky, R. D. Quinn. “An Obstacle-Edging Reflex for An Autonomous Lawnmower,” 2010 IEEE/ION Position Location and Navigation Symposium (2010 ION/IEEE PLANS), Indian Wells, CA, May 2010, pp. 74-79.

[10] A. Schepelmann, et al. **“**Vision-Based Obstacle Detection for the CWRU Cutter Autonomous Lawnmower,**”** In Proceedings of the IEEE International Conference on Technologies for Practical Robot Applications (2009 IEEE TePRA), Woburn, MA, Nov. 2009.

[11] R. Sheh, et al. “A Low-Cost, Compact, Lightweight 3D Range Sensor,” In Proceedings Australian Conference on Robotics and Automation, Auckland, New Zealand, 2006.

[12] S. Thrun *et al.*, "Stanley: The robot that won the DARPA grand challenge: Research articles," *J. Robot. Syst*, vol. 23, no. 9, pp. 661–692, 2006.

[13] P. Bellutta, R. Manduchi, L. Matthies, and K. Owens, "Terrain Perception for Demo III," *IEEE Intelligent Vehicle Symposium*, Dearborn, USA, October 2000.

[14] G. N. De Souza, A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237–267, 2002.

[15] S. McMichael. "Lane Detection for DEXTER, an Autonomous Robot, in the Urban Challenge." M.S. Thesis, Case Western Reserve University, Cleveland, OH, USA, 2008.

[16] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[17] J. C. Maxwell, "On the theory of compound colours, and the relations of the colours to the spectrum," *Phil. Trans. R. Soc*, 150, 1860, pp. 57-84.

[18] A.R. Smith. "Color gamut transform pairs". Computer Graphics 12(3): 12–19, 1978.

[19] GNU Image. "HSV Color Cone." Retrieved from <http://ilab.usc.edu/wiki/index.php/Image:HSV_Color_Cone.png>

[20] M. Zaheer Aziz, B. Mertsching, M. Salah E. N. Shafik, and R. Stemmer, "Evaluation of Visual Attention Models for Robots," Proceedings of the Fourth IEEE International Conference on Computer Vision Systems, 20, January 04-07, 2006

[21] J. G. Liu, J. M. Moore "Hue image RGB colour composition. A simple technique to suppress shadow and enhance spectral signature." International Journal of Remote Sensing, vol. 11, no. 8, 1990, pp. 1521 – 1530.

[22] I. Philipp and T. Rath, "Improving Plant Discrimination in Image Processing by Use of Different Colour Space Transformations," Computers and Electronics in Agriculture, 35, pp. 1-15, 2002.

[23] J. Hemming and T. Rath, "Computer-Vision-Based Weed Identification Under Field Conditions Using Controlled Lighting," J. Agricultural Engineering Research, 78(3), pp. 233-243.

[24] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2008.

[25] D. Marr and E. Hildreth, "Theory of edge detection," Proceedings of the Royal Society of London, B 207 (1980) 187–217.

[26] U. Watchareeruetai, et al, “Computer vision based methods for detecting weeds in lawns,” Machine Vision and Applications (2006) 17:287–296.

[27] A. Broggi, C. Caraffi, P. P. Porta, and P. Zani, “The Single Frame Stereo Vision System for Reliable Obstacle Detection used during the 2005 DARPA Grand Challenge on TerraMax,” Proc. Of the IEEE ITSC 2006, Sept 17-20,2006, pp. 745-752.

[28] G. Chen. “Texture Based Road Surface Detection 3D Morphological and Texture Analysis.” M.S. Thesis, Case Western Reserve University, Cleveland, OH, USA, 2008.

[29] B. Zafarifar and P. H. N. de With, “Grass Field Detection for TV Picture Quality Enhancement,” IEEE International Conference on Consumer Electronics, ICCE2008 , vol. p. 329-330, January 2008, Las Vegas.

[30] P.C. Mahalanobis, “On the generalised distance in statistics.” Proceedings of the National Institute of Sciences of India (1934) 2 (1): 49–55.

[31] GNU Image. “Chrysanthemum sp.” Retrieved from <http://en.wikipedia.org/wiki/File:Chrysanthemum_sp.jpg>