UNIVERSITY OF CALIFORNIA, SAN DIEGO

A Holistic Approach to Human Presence Detection On Man-Portable Military Ground Robots

A Thesis submitted in partial satisfaction of the requirements for the degree
Master of Science

in

Computer Science

by

Frederick Christopher Birchmore Jr.

Committee in charge:

Professor Serge Belongie, Chair
Professor Javier Movellan
Professor Lawrence Saul

2009

The Thesis of Frederick Christopher Birchmore Jr. is approved and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2009

DEDICATION

To my friends/colleagues at the Unmanned Systems Branch at the Space and Naval Warfare Systems Center, Pacific (SPAWAR) for the use of their robots and for their continued support.

EPIGRAPH

”Sit down before fact as a little child, be prepared to give up every preconceived notion, follow humbly wherever and to whatever abyss nature leads, or you shall learn nothing.” - Thomas Huxley

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

There are many people who I would like to thank for making this thesis possible: Dr. Serge Belongie for introducing me to computer vision, teaching me, and inspiring me to make a career out of it. Dr. Javier Movellan and Dr. Lawrence Saul for agreeing to be on my committee. Greg Kogut, my friend and colleague at SSC Pacific, for keeping me on the right track and for his good advice. Bart Everett, Technical Director for Robotics at SSC Pacific, for mentoring me and for reminding me to never lose sight of the big picture. The Science, Mathematics And Research for Transformation (SMART) Scholarship for Service Program for funding my graduate education. Dr. Jim Rohr for introducing me to SSC Pacific. My other friends/colleagues at SSC Pacific. My Dad and Aunt Linda for being so supportive throughout my education. The Grossmont-Cuyamaca Community College system – without which I might never have made it this far. The U.S. armed forces for keeping us safe at home and abroad. Everyone else I haven't mentioned who helped make this possible. Most importantly, I would like to thank God who makes all things possible.

ABSTRACT OF THE THESIS

A Holistic Approach to Human Presence Detection On Man-Portable Military Ground Robots

by

Frederick Christopher Birchmore Jr.

Master of Science in Computer Science

University of California, San Diego, 2009

Professor Serge Belongie, Chair

Object detection is a well-explored problem in computer vision. Although object detection algorithms have improved greatly in terms of speed and accuracy, many of these algorithms cannot be used directly on man-portable ground robots in-theatre due to poor detection rates and/or runtime efficiency when exposed to the low-quality camera data and processing limitations of these robots. In order to facilitate the introduction of near-term practical object detection capabilities to these robots, this thesis argues that a ”holistic” approach should be taken which views the robot as a system with its vision being only a small part of its senses. This thesis analyzes many different aspects of man-portable ground robots from the history of their use to technological and philosophical obstacles to deploying these robots with autonomous behaviors. To demonstrate this ”holistic” approach, this thesis introduces a human presence detection (HPD) method which utilizes the robot's ability to physically maneuver itself which will hopefully lead to better detection rates. This is simulated using difficult multi-scale testing data designed to expose the strengths and weaknesses of the HPD method in this thesis and to pave the way for future development. It is hoped that further integration of the robot's behaviors, the use of a multitude of different sensors, and improved interaction with human soldiers will lead to object detection capabilities (such as HPD) which will be deployable on man-portable ground robots currently used by soldiers in-theater.

Chapter 1

Introduction

1.1 Problem

The longterm goal behind this thesis is to supplement and ultimately replace the human soldier on the battlefield with a robot. Human pilots in-theater are already being replaced to some extent by UAVs (Unmanned Aerial Vehicles), such as the *Global Hawk* and the *Predator*, with the *Predator* even being armed with *Hellfire* missiles [63; 64]. Ground robots, such as the *PackBot* and the *Talon*, are currently being used on the battlefield but they are relatively far from replacing any human soldiers [30; 23].

The ground robots currently deployed in-theater actually require several soldiers for operation. Typically, a soldier who has been trained to operate a specific type of robot must give his/her undivided attention to an OCU (operator control unit), such as a laptop, while several other soldiers protect him/her from dangers such as enemy sniper fire. Thus, rather than replacing the human soldier on the battlefield, ground robots currently require an increase in the number of human soldiers on the battlefield.

It could be argued that technology simply has not yet caught up to the requirements that ground robots have and that, given time, the appropriate technology will exist to increase the robot's autonomy and thus decrease the robot's reliance on human operators. Ground robots, however, have actually been deployed on the battlefield since World War Two when the *Goliath Tracked Mine* was first introduced [18]. Now, almost seventy years later, ground robots in-theater have remained fundamentally unchanged while UAVs have become extremely prevalent – especially in the form of guided missiles which have also been around since the introduction of the World War Two *V2* rocket , which is considered to be the first deployed autonomous vehicle [18].

Although this thesis does not purport to resolve the longterm goal of replacing the human soldier with a robot, it is hoped that this thesis will help contribute to a near-term, practical introduction of autonomous behaviors to mobile man-portable military ground robots. Within this near-term goal of introducing autonomous behaviors, this thesis will specifically focus on the object detection task of detecting people, commonly referred to as human presence detection (HPD). Rather than relying on the data from a single sensor, such as images from a camera, this thesis argues that autonomous behaviors should be addressed ”holistically” by treating the robot as a system with many different components working together to interact with its environment. Specifically, the HPD algorithm in this thesis aims to combine the robot’s ability to physically reposition itself with data from an infrared camera in hopes of achieving better detection results than using image data alone could provide. By addressing robot behaviors holistically in general, it is hoped that behaviors which otherwise would not be mature enough or practical enough to use on ground robots currently deployed in-theater will be promoted to a deployable status in the near future.

1.2 Why Is Object Detection Difficult?

Consider an 8x8 binary image. If a database were created that contained all possible 8x8 images, it would contain 2^{64} or 1.8447×10^{19} entries. The images of a specific object under all possible rotations, scales, and lighting conditions can be represented by a finite number of these combinations. In the broadest terms, all of the solutions to detecting an object using an 8x8 binary image lie within a subset of these possibilities. Many objects, however, are not detectable with an 8x8 binary image. If a 320x240 8-bit grayscale image were used to create the database, the number of images would rise to $6.7505 \times 10^{184952}$, as shown in Equation (1.2). Hypothetically, even if this were computed on the world's most powerful supercomputer, the *Roadrunner* at the Los Alamos National Laboratory, running at one petaflop per second (see Equation (1.1)) and each image comparison took only a single floating point operation, then it would take $2.1406 \times 10^{184930}$ years to match a single image through brute-force in the worst-case scenario, as shown in Equation (1.3) [41]. To put this number into perspective, if we live in an open universe, it is expected that every star in the universe will burn out and all black holes will evaporate in 10^{100} years [3]. Even if the supercomputer survived, it would still have 10^{184830} years to go until it completed its search of the database.

$$1\, petaflop = 10^{15}\ flops\ per\ second \tag{1.1}$$

$$256^{320\times240} = 6.7505 \times 10^{184952}\ images \tag{1.2}$$

$$\frac{6.7505 \times 10^{184952}}{10^{15} \times 60 \times 60 \times 24 \times 365} = 2.1406 \times 10^{184930}\ years \tag{1.3}$$

Fortunately, there is no reason to store all possible images in a database, especially if only a single object or a subset of objects to detect are stored. If all possible images of just a single object to be detected could be stored, then the size of the database would obviously decrease dramatically. The trick to this approach, however, would be to acquire images of an object under all possible conditions which would still result in many images. Many of these possibilities, however, could be eliminated if the scope of the problem were narrowed further. At a high level, this is how object detection methods work. For example, if the object to be detected is known to be perfectly square, then a filter could be run on the image to detect only perfectly vertical and horizontal lines. In this case, a database of all the possible images could exist *implicitly* within the space of all possible filter responses and the number of possibilities would therefore be much smaller than storing a set of all possible images of the object.

In this thesis, the core of the HPD problem is based on the assumption that the temperature of humans will fall within a certain range and that other objects in the scene will not take on this temperature range as readily. The images are thresholded to this temperature range, eliminating many different possibilities. In a cool, indoor environment, all of the images that contain pixel values tend to contain humans – effectively reducing the problem space to contain only the solutions. The HPD algorithm would not be practical, however, if it were restricted to cool indoor environments only. In order to extend this algorithm towards practical situations, there must be some sort of transition point towards a practical solution. This transition point starts with outdoor images of buildings and involves working to separate them from the cool indoor images of humans.

1.3 Human Presence Detection Defined

HPD is effectively a more specific form of object detection. It happens to be one which we humans are extremely good at since we see and interact with other humans all the time in our everyday life. We detect the presence of humans by appearance, smell, and motion. In fact, computer animators have an especially difficult time animating human motion because humans are so good at detecting human characteristics that the illusion is often spoiled by subtle differences in movement, appearance, and mannerisms, as mentioned by Hodgins, et al [29]. Similarly, however, older science-fiction movies had trouble getting humans dressed up in monster costumes to look more like monsters and mask the ”it’s just some guy (or girl) in a suit” impression. The fact that humans are so distinct in their behaviors can be exploited by a computer to perform HPD potentially much easier than general object detection.

1.4 What Makes a Human Detectable?

There are many different aspects of humans that can be exploited for detection. First of all, (live) humans are warm blooded and the body temperature of a healthy human is generally a constant 98.6 degrees Fahrenheit. Another aspect of humans is the presence of low-frequency curves, such as the round shape of their head and shoulders. Human-made structures tend to have high-frequency edges so perhaps high-frequency edges could be pruned in some way. If a human is moving, their gait pattern could be exploited for detection. The human gait pattern is relatively constant. The sound of human footsteps could also be used to indicate the presence of a human, as was done by Damarla [13]. The most common method for detecting humans, however, is with visual and/or infrared images.

1.5 Previous Work

There is plenty of literature for HPD, much of which involves the use or development of image databases shared within the computer vision community [16; 49; 43; 12; 74; 71]. Dalal and Triggs used *Histograms of Oriented Gradients (HOG)* to perform HPD using a database of pedestrians [12]. Almageed and Davis used edge maps to perform HPD on a database as well [2]. Zhang, et al combine *Edgelets* with *HOG* features to detect pedestrians in an infrared database [12; 71; 74].

Methods fusing multiple data sources together have recently emerged, often combining infrared images with color images and/or stereo depth maps. Some systems employ infrared and color images from multiple stereo pairs (i.e. tetravision-based detectors) [8; 39].

HPD on robots or other vehicles is also quite common. Spinello and Siegwart combined laser range data with *HOG* features from camera images to detect pedestrians using data from a moving automobile [12; 55]. Rudol and Doherty perform HPD in real-time from a UAV (Unmanned Aerial Vehicle) by combining infrared and color imagery and using a cascade of boosted classifiers trained using Haar-like features [48; 24; 70].

To the author's knowledge, the specific process in this thesis of considering the robot's motion to influence HPD results has not been published before. This idea was mentioned by Kogut, et al though it was not tested [27]. The robot's motion combined with its sensors is typically used for tasks such as navigation, obstacle avoidance, and mapping. A well-known example of utilizing the robot's motion to navigate is the *SLAM* algorithm *(Simultaneous Localization And Mapping)* which is used by robots to build up a map of an unknown environment while at the same time keeping track of their current position [52; 4]. The approach taken in this thesis, compared to other research, is to explore the obstacles the robot has to becoming a deployable system by exploring its ability to compensate for its weaknesses in some areas by utilizing capabilities in other areas.

1.6 Thesis Structure

This thesis is organized into nine chapters. Chapter 1 is the introduction. Chapter 2 provides a historical background of man-portable ground robots in the military and describes the various roles they are currently filling. Chapter 3 describes the problem of transferring technology from research to deployment and offers philosophical solutions. Chapter 4 describes an approach toward providing man-portable ground robots with autonomous behaviors by treating computer vision as just one portion of a robot's sensory system. Chapter 5 describes the target robotic platform for which the experiments in this thesis are developed, the sensors available, its behaviors, and which of these capabilities are exploited for Human Presence Detection (HPD) in this thesis. Chapter 6 describes in detail the methods employed in this thesis for the visual HPD portion, including the training data acquired. Chapter 7 describes in detail the methods employed in this thesis for the physical HPD portion, including the testing data acquired to simulate execution on a robot. Chapter 8 describes the experiments performed to evaluate the effectiveness of the HPD methods in this thesis and tests its limitations. Chapter 9 summarizes this thesis, draws conclusions from the results, and specifies opportunities for future work.

Chapter 2

History of Man-Portable Ground Robots in the Military

2.1 Man-Portable Robots in the Second World War

One of the earliest incarnations of a man-portable ground robot used on the battlefield was the German-made *Goliath Tracked Mine* as seen in Figures 2.1, 2.2, and 2.3 [18]. *Goliaths* were used by the German armed forces (the Wehrmacht) on all fronts beginning in spring 1942 during World War Two [18]. Each tracked vehicle was approximately four feet long, two feet wide, and one foot tall [18]. The *Goliaths* were designed to carry a 60kg high-explosive charge which would be detonated when the *Goliath* was driven under a tank, destroying both the tank and the *Goliath* vehicle [18]. The vehicle was controlled via a joystick and a 3-strand telephone cable connected to the rear of the vehicle [66; 18]. *Goliaths* were man-portable in that when they weren't running under their own power, they could be carried by hand on a trailer pulled by two soldiers, as seen in Figure 2.3 [68].

Although a total of 7,579 *Goliaths* were produced, they were not considered a success due to their high unit cost (300 Reichmark), poor ground clearance, low speed (slightly above 6 miles per hour), their thin armor, and their vulnerable cable which

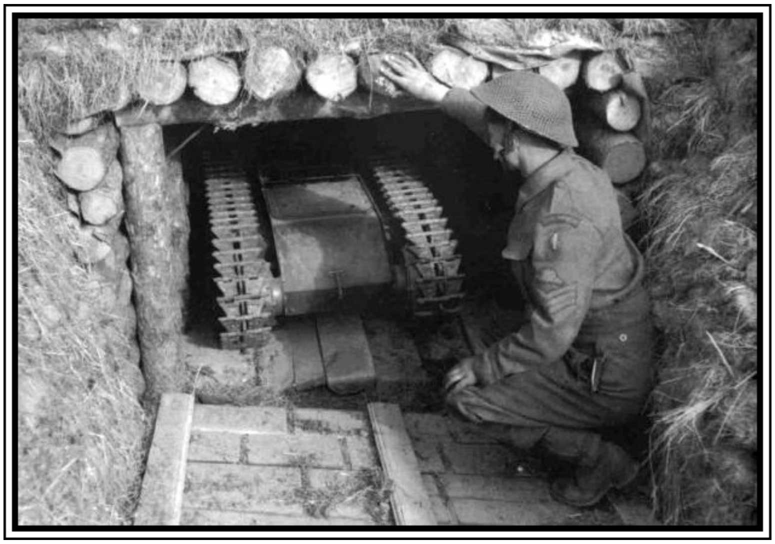

Figure 2.1: A British soldier with a *Goliath Tracked Mine*.[67]

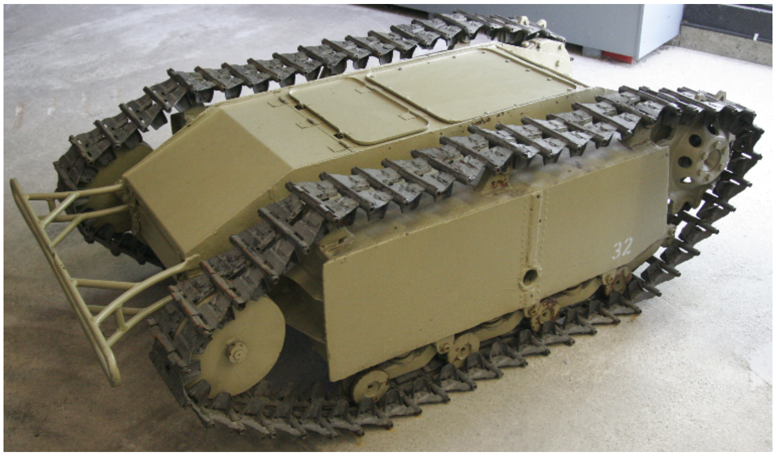

Figure 2.2: A modern-day photo of a *Goliath Tracked Mine*.[7]

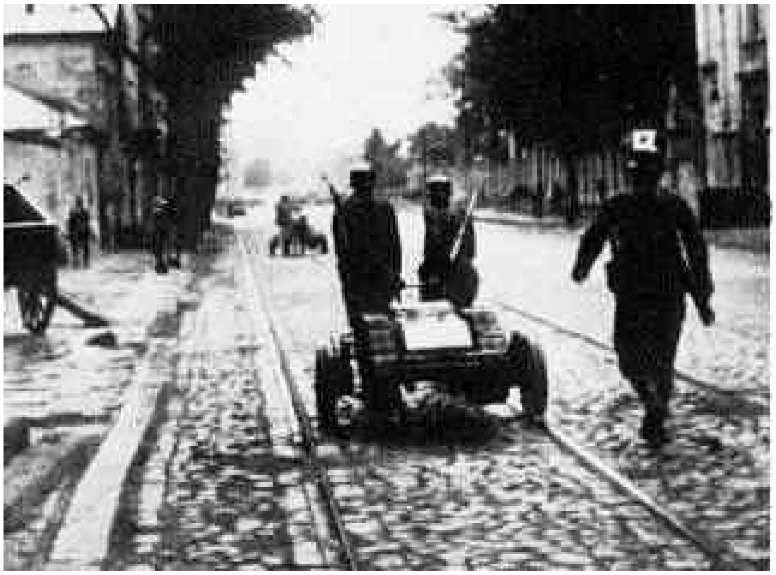

Figure 2.3: German soldiers towing a *Goliath* with a trailer in Warsaw, 1944 [68].

could be easily severed [66; 18].

2.2 Man-Portable Robots in-Theatre Today

2.2.1 *PackBot* and *Talon*

The two robotic platforms most widely used in-theatre today are iRobot's *PackBot* as shown in Figure 2.4 and Foster-Miller's *Talon* as shown in Figure 2.5 [30; 23]. The *PackBot* can be controlled using a wireless communications link but this is only good up to 3280 feet while the robot is within line-of-sight [31]. A tether can also be attached to achieve a wired-link. Its physical dimensions are 20.5 inches wide, 34.6 inches long, and 7.5 inches high [31]. It weighs around 42 pounds and can operate in all weather conditions [31].

The *Talon* can be controlled using an operator control unit which by itself weighs 33 pounds and has physical dimensions of 16 inches wide, 19 inches long, and 9 inches high [23]. The *Talon* can be controlled wirelessly up to 4000 feet while the robot is

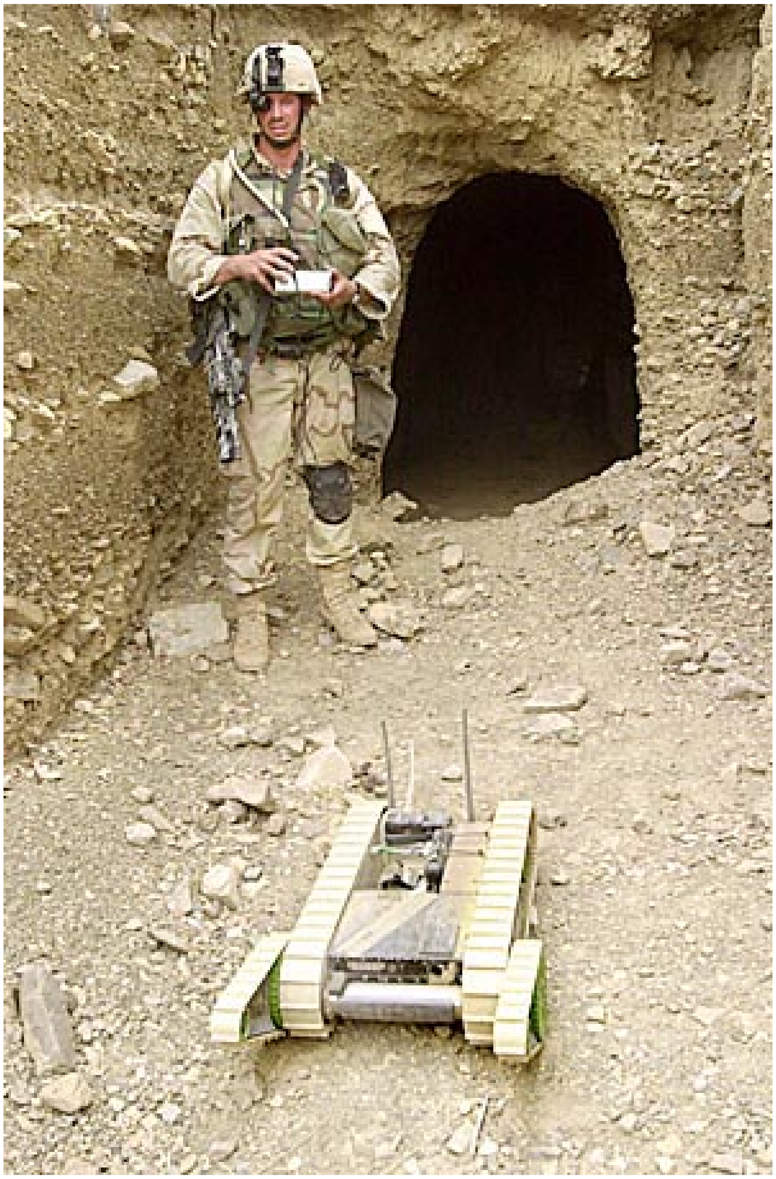

Figure 2.4: ”A U.S. soldier maneuvers a *PackBot* robot named Hermes in front of a cave to detect mines and other unexploded ordnance as well as weapons or equipment possibly hidden by Taliban or Al-Qaida fugitives near the eastern border town of Qiqay, Afghanistan [36]”.

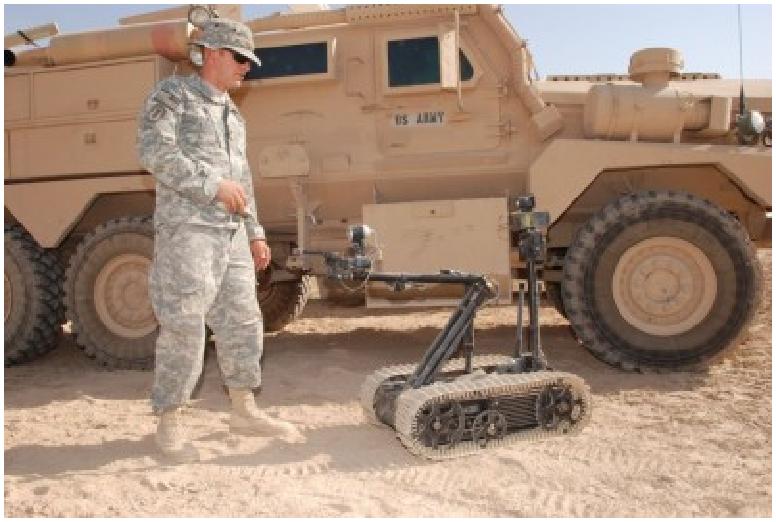

Figure 2.5: Spc. Dennis Speek, explosive ordnance disposal technician, 789th Ordnance Company (EOD), explains how EOD personnel use the *Talon* robot to handle explosives by remote control [57].

within line-of-sight of the operator control unit [23]. The *Talon* weighs between 115 and 156 pounds depending on which accessories are added to it and has physical dimensions of 22.5 inches wide, 34 inches long, and 18 inches tall with an attached arm [23]. It is capable of operating under all weather conditions [23].

Almost seventy years have passed since the *Goliath* robot was first used [18]. Although man-portable robots have improved tremendously in many ways since then, **in many ways they have not changed at all**. Like the *Goliath* robots, today's man-portable robots are remotely operated using a joystick. They have limited range and often require a physical tether to operate in areas where radio communication links cannot be maintained. Probably the largest improvements today's robots have over the *Goliath* is the ability to be carried rather than pushed in a cart and the ability to send data back to the operator - like the television-guided missiles of 1941 [18].

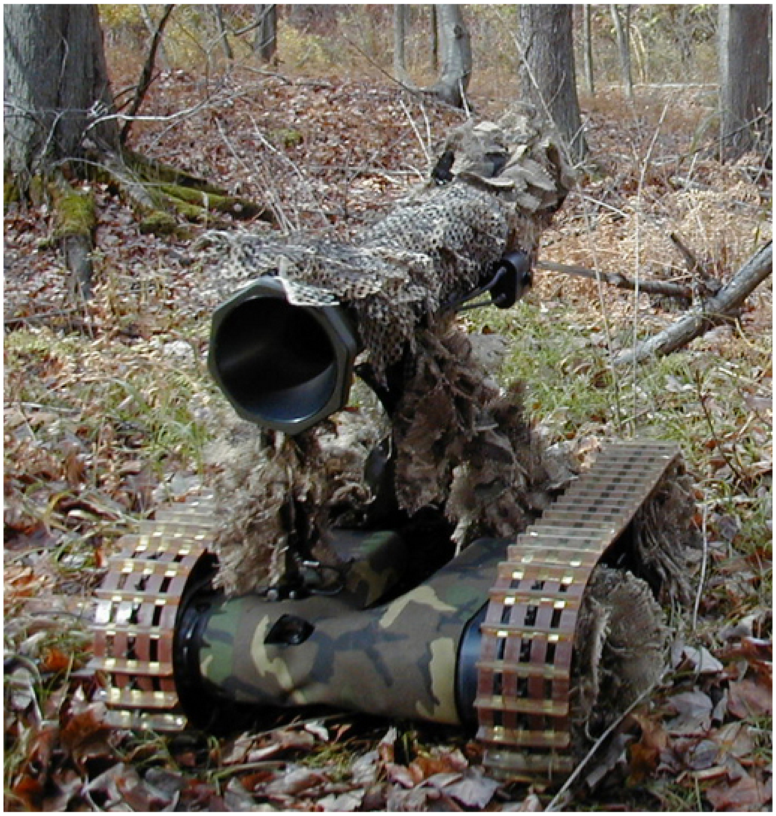

Figure 2.6: A *SWORDS* variant of the *Talon* man-portable robot with camouflage [14].

2.2.2 Robots as Combatants

Most enemy combatants value their lives and would do everything in their power to avoid defeat and achieve victory for themselves. Some of today's man-portable robots are armed with weapons, such as the *SWORDS* variant of the *Talon* robot, but the effectiveness of these robots in combat would be very limited [23]. *Talon SWORDS* robots can be configured with M240 or M249 machine guns or Barrett .50 caliber rifles which gives them considerable firepower [51]. When compared to a human combatant, however, who is is faster, quieter, seldom runs low on power, and can blend in with local populations, a robot is no match. The only chance current robots have on the battlefield is to fill very specific niches. According to Defense Review:

"The key to the survivability of *SWORDS* would appear to depend on the remote operator's ability to successfully 'sneak and peek' with *SWORDS* Unmanned Ground Vehicle (UGV) to locate and shoot/kill the enemy before the enemy locates and shoots/kills *SWORDS*, then exfiltrate the area as quickly as possible or look/patrol for the next enemy target(s). Low observability/low visible signature would thus seem to be a necessary combat attribute for *SWORDS*. (see Figure 2.6) [14]"

Both the *Talon* and the *PackBot* are quite loud and enemies would be able to hear them coming from short distances, which amplifies the problem of stealth. In combat, today's man-portable robots would easily be defeated by enemy combatants. As robotic technology advances, robots will gradually fill an increasing number of niches on the battlefield.

2.2.3 Robots as Non-Combatants

Currently, the main advantage man-portable robots have over human soldiers is their expendability, size, and ability to withstand conditions which humans cannot. Man-portable robots have performed very well when used by Explosive Ordnance Disposal teams (EOD) to disarm explosive devices [54]. In the presence of hazardous materials, such as nuclear, chemical or biological contaminants, man-portable robots can go places where humans cannot. In the wake of a natural or man-made disaster, man-portable robots can navigate rubble and venture into narrow passage-ways into which a human would not be able to fit. The role of man-portable robots is still rather limited in situations like these in terms of the level of human supervision required to control the robot and the loss of comm-signals between the robot and the OCU. If an EOD team member is operating the controls of a robot to disarm an explosive device, he/she requires an extra amount of focus to operate the OCU in addition to disarming the explosive device. This leaves the operator susceptible to enemy sniper fire. Many of these problems could be reduced or eliminated altogether if man-portable robots could be given autonomous behaviors.

Chapter 3

Bridging The Gap Between Theory and Application

3.1 A Science-Fiction Example

It is often thought that research and industry work together in a very ideal manner. To illustrate this concept, an allusion to the fictional scientist ”Doc Brown” from the movie *Back to the Future* can be drawn, as shown in Figure 3.1 [73]. He invented the ”Flux Capacitor” which, according to him, ”is what makes time travel possible” [73]. If Doc Brown were to research, develop, and deploy his invention, he would ideally go through the following steps:

1. Fall down while fixing a toilet and bump your head. Imagine the Flux Capacitor once consciousness is regained.

2. Build a working prototype for a Flux Capacitor which never fails as long as it receives a healthy supply of weapons-grade plutonium.

3. Give blueprints to a factory which will begin mass-producing Flux Capacitors.

Although some inventions are developed in this manner, the vast majority are not. If it is hypothetically assumed that a Flux Capacitor is possible, a more realistic set of steps

Figure 3.1: Doc Brown (played by actor Christopher Lloyd) in a screenshot from the movie *Back to the Future* holding a schematic of the Flux Capacitor which he came up with after hitting his head on the toilet [73].

might proceed as follows:

1. After finishing 24 years of schooling, you realize how difficult it is to get a job with a degree in Applied Theoretical Physics. You reason that the ideal solution to this problem would be to travel back in time and convince one's self not to pursue such a degree. With such an extensive background in physics, and the basic theoretical principals of storing "Flux" to travel through time, you start a "Flux Capacitance Theory" lab at a prestigious university.

2. Your lab develops countless theorems and controlled laboratory experiments where small sub-atomic particles are sent back in time.

3. After years of research and publications, another lab demonstrates sending entire molecules back in time.

4. Years later, it is assumed that with the current model of time-travel, it is not possible to send multiple molecules back in time while keeping their molecular bonds intact.

5. You become elderly, retire, and pass away.

6. Years later, another person comes across a publication on your original theory of ”Flux Capacitance”, thinks about it in a different way, and figures out how to send molecules back in time while still preserving molecular bonds.

7. Years later, a clever entrepreneur purchases this technology to start a drug company which uses this technology to send medicine back in time to cure dying people, effectively ”raising the dead”.

Although it is important to come up with original ideas, these ideas may not be feasible for many years and their ultimate utility may manifest itself in an entirely different way. When some people begin a project, they often have the misconception that their project will ultimately follow the steps in the first example from the movie. When the results of that project fail to meet those expectations, disappointment ultimately ensues. The result from a project that tries too hard to meet those expectations is often a loose estimate of those expectations but fails to exhibit any practical utility. There have been many important projects in the defense industry over the years which have seen deployment, but a vast majority have not.

In research, cutting-edge technology is often improved by researchers who then publish their findings. These findings are then incrementally improved by these researchers or others. Research, industry, and the military all have different requirements for the usefulness of inventions. Sometimes, the newest research findings can be directly applied in industry or the military and perform flawlessly. Most of the time, however, this is not the case. The discovery of a new element in the periodic table would potentially help other researchers discover further elements but this would most likely not help the pharmaceutical industry market new drugs or the military create more stable explosive compounds. For example, a new line-finding algorithm developed in research wouldn’t mean that automobiles could then autonomously stay within the painted lines separating lanes on the freeway. The researched technology would need to be integrated, for example, into an automobile with a drive-by-wire system and a camera. If this were accomplished successfully in industry, it would not mean that the military could then allow their vehicles to drive autonomously on roads. A military vehicle would potentially have to navigate dirt roads with no lines marking the lanes or an enemy force could deliberately paint over the lines on their roads to foil autonomous line-finding algorithms. At the other end of the spectrum, technology developed for the military cannot necessarily be used in industry. For example, if a new type of plastic explosive were developed which was more effective at breaking up rocks than the current state-of-the-art method, industry would not necessarily be able to use it for mining minerals if the plastic explosive were carcinogenic and required specialized training to handle safely.

3.2 A Real Example: The Manhattan Project

The gap between research, industry, and the military is sometimes narrow or can be forced to become narrower due to different circumstances arising. A good example of this is the Manhattan Project, the result of which yielded the world's first atomic weapon which was tested on July 16, 1945 [65]. It was only two years before the 1938 project began, in 1936, when it was discovered that unlocking atomic energy was even possible [65]. When the project ended, two working atomic bombs were put onto existing airplanes and successfully dropped on Japan. This situation is far from typical, though, and the extenuating circumstances which made this tremendous amount of work possible were a desire to save the free world.

3.3 How Research Is Advanced

Much of research comes from taking ideas that are used in one application or field and applying them to solve a different application. Every idea we form is inspired by something else and is often from something found in nature. The airplane, for example, was inspired by birds. For thousands of years, human-kind had watched birds soar effortlessly through the sky and wished they could do the same. When one sees a propeller airplane soaring through the sky, it often soars much like a bird.

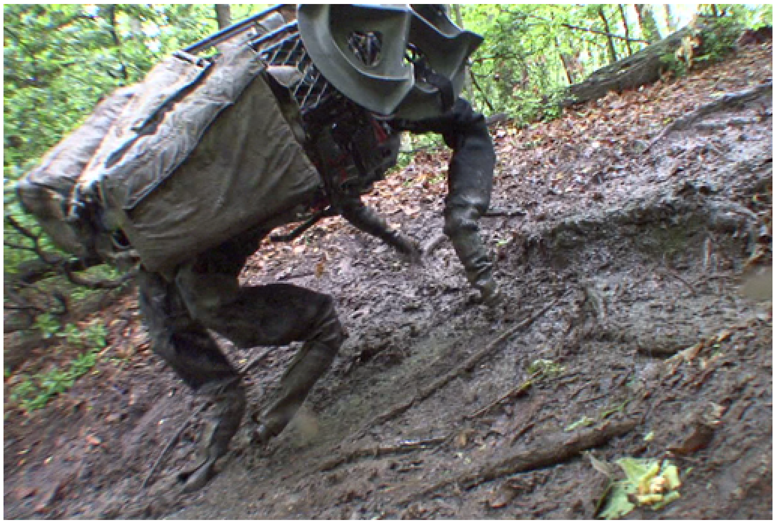

Figure 3.2: This image, taken from the Boston Dynamics website, shows their robot *BigDog* climbing a muddy slope [9].

As robots advance in technology, they will gradually begin to resemble humans or animals in many ways. A recent example of this is the *BigDog* robot developed by Boston Dynamics [9]. The legs of *BigDog* are articulated like a quadruped animal and can walk, climb, and carry heavy loads over rough terrain [9]. When it moves about carrying a heavy load, it often reminds people of two soldiers carrying a heavy load between them, as seen in Figure 3.2. When it stumbles to regain its balance, this image is enhanced.

The same research is often simultaneously repeated by many different people who may or may not have knowledge of what the others are doing. Sometimes, the same problem is actually solved simultaneously by multiple people who may or may not have knowledge of what the others are researching. This can be quite inefficient but is unavoidable. Particularly in the military, where sensitive information is understandably provided on a need-to-know basis, some R&D departments may be researching the same technology that other departments are already researching or already possess solutions to. There is a huge trade-off between publishing results and keeping them proprietary. On one hand, if results are published and made public, someone else might unfairly benefit from them. For example, if billions of dollars are spent on a missile defense system and all the research is published, then the means to both construct and potentially defeat the missile defense system would effectively be given to enemy nations – at no cost to these nations. On the other hand, if results are not published, then the same technology could be simultaneously invented by several people who would never know about each other's achievements. Many times, repeating research without the knowledge that the problem might had already been solved can be a benefit. For example, effective human presence detection algorithms might already exist on military hardware but the cost of such algorithms may be prohibitive for use on other hardware. The secrecy of the hypothetical hardware's technology could be essential for national security. In this case, independently developing a HPD algorithm without access to expensive hardware and without knowledge of the said hardware might yield a different HPD algorithm that is very affordable without jeopardizing national security.

In terms of research, the HPD algorithms discussed in this paper fall somewhere in-between. The hardware used in this thesis is not necessarily available to the general public due to the cost of the infrared camera, for instance. As time goes by, infrared cameras will undoubtedly drop in price and will most probably be replaced by better technology down-the-road. In the meantime, the cost of infrared cameras is cheap with regards to the cost of fighting a war and expendable with respect to the human cost of soldiers fighting in a war. Although this is frequently mentioned, its importance cannot be emphasized enough: a robot is expendable but a human soldier is not. Progress has hopefully been made by this thesis toward solving the ongoing problem of providing autonomous behaviors to robots. This is not the final solution to HPD but hopefully it will help point further research in the right direction.

Since technology is developed by so many different entities, prerequisites may or may not be met. One research problem may be dependent on the affordability of another technology which, in turn, may depend on the affordability of another technology. This, however, may not be apparent. If an object detection algorithm is very effective in terms of accuracy but is too slow to run in real-time, it may be dependent on waiting for more computing power or replacing an inefficient portion of that algorithm with something more efficient. One way of viewing this is by considering that there is a solution to every research problem but the difficulty lies in finding that solution. This idea has been formalized with the Theory of Inventive Problem Solving, known as TRIZ.

3.4 Theory of Inventive Problem Solving (TRIZ)

3.4.1 Introduction

TRIZ is a Russian acronym which translates to ”Theory of Inventive Problem Solving” [60]. TRIZ was developed by a Soviet engineer and researcher, Genrich Altshuller, and his colleagues beginning in 1946 and has been evolving ever since [60]. In contrast to brainstorming, which is essentially based on random idea generation, TRIZ is a theory and methodology that aims to create an algorithmic approach to inventing new systems and refining existing systems [44]. TRIZ has been developed by commercial promoters into a collage of concepts and tools. The main reason for introducing TRIZ in this thesis is not to advocate TRIZ specifically as a method for inventing, but rather as a means of fortifying the philosophy behind developing object detection capabilities in this thesis so that the work presented in this thesis and future work in this area can help push mobile man-portable ground robots toward becoming deployable, autonomous machines.

3.4.2 Origin of TRIZ

Altshuller developed TRIZ by screening over 200,000 patents for inventive problems and how they were solved. He classified the patents according to the inventiveness of the solution [28]. The solutions were categorized into five levels, as shown in Table 3.1.

Altshuller found that over 90% of the problems engineers faced had been solved somewhere before [28]. If engineers could follow a path to an ideal solution, starting

Level	Degree of Inventiveness	% of solutions	Source of knowledge	Approximate # of solutions to consider
1	Apparent solution	32%	Personal knowledge	10
2	Minor improvement	45%	Knowledge within company	100
3	Major improvement	18%	Knowledge within the industry	1000
4	New concept	4%	Knowledge outside the industry	100,000
5	Discovery	1%	All that is knowable	1,000,000

Table 3.1: Altshuller's findings from screening patents.[28]

with their personal knowledge and experience and working their way up through the levels shown in the table, most of the solutions could be derived from knowledge already present in the company, industry, or some other industry [28]. Altshuller distilled the problems and solutions he found in these patents into a theory of inventive problem solving which he named TRIZ.

The main idea behind TRIZ is that invention is the process of removing technical contradictions, according to Altshuller. An example of a so-called technical contradiction is ”If we want more acceleration, we need a larger engine; but that will increase the cost of the car” [59]. For object detection in mobile man-portable ground robots, an increase in the detection rate often translates to a requirement of more computational power. More computational power can be introduced by putting more powerful processors on the robots, but this leads to more power requirements, such as heavier batteries. So much of the robot's resources are dedicated to other higher-priority processes such as mobility and communication links that adding some object detection algorithms could cause more harm than good. Faster detection requires either more processing power or more efficient algorithms which often achieve lower detection rates.

In order to introduce practical object detection capabilities to mobile man-portable ground robots, efficient HPD methods should be sought but utilizing other behaviors and sensors on the robot should compensate for their lower detection rates. The robot's ability to move and modify its behavior should be exploited so that lower detection rates would not interfere with the robot's mission. This thesis is advocating the idea that if object detection is to be used at all, it should not interfere with other more essential processes of the robot, such as its ability to communicate and to move.

3.5 A TRIZ-like Approach to Object Detection for Mobile Man-Portable Robots

Principals described with TRIZ explain that one contradictory parameter for a problem is typically traded for another and that no special inventiveness is needed for this. The inventor aims to develop some creative approach for resolving the contradiction, such as inventing an engine that produces more acceleration without increasing the cost of the engine. For mobile robots, the ideal goal should be to introduce full-autonomy to the robots, but the interim goal should be to introduce object detection capabilities which, when added to the robot, would not demote the robot from ”deployable” to ”research platform”. By keeping this as a strict requirement and yet not ruling out object detection as a feasible solution, this sets other parameters as flexible. Some ”control” parameters, as in ”fixed” parameters, to follow for this thesis are:

1. The introduction of object detection capabilities to a deployable robotic platform should not demote its status from ”deployable” to ”research platform”.

2. The object detection capabilities must provide a net benefit for the robot that is enough to justify adding these capabilities to the robot.

3. The object detection capabilities must not significantly interfere with the robot's (nor human soldier operator's) ability to perform tasks of greater importance, such as the robot's ability to move.

By setting these control parameters in place, other "variable" parameters which are often overlooked or sidelined in object detection research now become adjustable:

1. Detection rate with respect to the utility of the object detection algorithm
2. Detection rate with respect to the runtime and depth of data analysis
3. Robustness to new data with respect to the specificity of the algorithm (which is often inversely proportional).

3.5.1 Detection rate with respect to the utility of the object detection algorithm

If the detection rate of a hypothetical object detection algorithm is between 65% and 95% for true negatives but 100% for true positives, then the algorithm should be restricted to tasks where this detection rate is acceptable. At the same time, the algorithm must still be useful enough to justify putting it on the robot. A true negative detection rate of 65% is not be acceptable if the robot's task is to detect and then shoot enemy threats, because the false negatives could represent friendly soldiers, resulting in the loss of soldiers' lives as a result of the robot's error. It might be acceptable to have a true negative detection rate of 65%, however, if the robot's task is to display objects of interest on a map which will later be verified by a human. This would ideally still result in removing some of the work that a human would otherwise need to perform –which would result in adhering to control parameters 1 and 2. On the other hand, if in the process of achieving a detection rate of 65%, extra work is created for the human who is checking the map that does not create a net benefit for the robot or the other humans, then either the detection rate should be increased or other parameters of the robot or map should be modified to maintain a net benefit and justify adding the object detection method.

3.5.2 Detection rate with respect to the runtime and depth of data analysis

Often in object detection research, a higher detection rate requires a more computationally complex algorithm which can result in a longer run time. More efficient object detection algorithms often use simpler algorithms which use rougher approximations, resulting in a lower detection rate.

The most efficient object detection algorithms tend to perform less detailed analysis of the data or use less of it. For example, the Haar-like features used by Viola and Jones for detecting faces analyze different configurations of pixels by performing simple statistical methods on the sums of pixels within rectangular regions of greyscale images [70]. In this case, the color data is in effect averaged once color images are converted to greyscale – abstracting the different color channels. Furthermore, the individual pixels within regions of the Haar-like features are abstracted into sums. The mathematical difference between these sums is the only portion of the pixel data that is saved. The weighted sum of these differences is used to identify a particular region of the image as being a face or not by comparing it to statistics gathered on the weighted sums extracted from other face and non-face images. The main reason this algorithm performs face detection efficiently is that it effectively skips most of the information in the image data. Its good detection rate is due to its ability to cleverly abstract the image data to make use of the most essential portions.

Computers are very good at performing mathematical operations such as adding, subtracting, and comparing equality. They are not very good at human tasks, however, such as detecting objects. It naturally makes sense that computers are capable of detecting objects once they are translated into a way which makes mathematical sense to the computer. Once an image is abstracted into numerical pixel values, the computer can apply math to it. Since the world is constantly changing and the pixel configuration of images of a particular object is highly variable, computers applying statistics and probability techniques to these images makes sense because these forms of math allow for a large amount of flexibility.

Many object detection algorithms take image data and apply many different complex operations to abstract it in different ways. Rather than taking image data and applying complex methods to abstract it, data from different sensors can be combined. For example, a depth map from a stereo camera could be combined with an image from a regular camera to help abstract the image data into separate objects or surfaces. Data from a laser range-finder (LIDAR, meaning Light Detection And Ranging) could then be used to separate correct image depth values from the incorrect ones. Combining techniques such as these could effectively isolate the important parts of the image to reduce the search-space for the object detection algorithm. This could possibly allow more complex object detection algorithms to be run in a reasonable amount of time.

If an object detection algorithm's runtime cannot be reduced to a reasonable amount of time such that the third control parameter is intact, then the context in which the algorithm is run could be modified. For example, if the object detection algorithm runs at seven frames per second but the physical speed of the robot is slowed down too much, then the algorithm could be run on every seventh frame to achieve a runtime speedup of 7x. The robot's behavior could be also be modified such that only non-overlapping frames are checked. The robot's search pattern could be modified such that objects will only be detected when the robot is within 10 feet of a wall, determined using LIDAR range values. This would hopefully allow the third control parameter to be upheld: ”The object detection capabilities must not significantly interfere with the robot's (nor human soldier operator's) ability to perform tasks of greater importance, such as the robot's ability to move”.

3.5.3 Robustness to new data with respect to the specificity of the algorithm (which is often inversely proportional).

If an object detection algorithm specifically searched for the brand name ”Cherrios” to detect a ”box of cereal”, then the algorithm would not be robust to any cereal that has a different brand name. This algorithm would be very specific in that boxes of Cheerios would be distinguished from other cereal boxes but not robust in that other boxes of cereal would not be detected and Cheerio boxes which are not oriented to display the word ”Cheerios” would not be detected. On the other hand, if an object detection algorithm detected cereal boxes by searching for sets of parallel lines, then that algorithm would not be robust to data which has parallel lines in it but no cereal boxes. Hence, a balance must be struck between how robust and how specific an object detection algorithm is.

3.6 The Small Robot Technology Transfer Program

The idea that technical problems in one field may have been solved in other fields sits at the core of the Small Robot Technology Transfer Program at the Unmanned Systems Branch at SSC Pacific [46]. The purpose of this program is to ”extract relevant aspects of various research activities, port them to related projects, and foster emergent technology transfer opportunities” [46]. This involves tracking technological developments throughout universities and industry, identifying relevant technologies, and developing them to work on robots. With the Technology Transfer Program, behaviors can be developed for the robots which could be modified as better technology becomes available. Having access to various technologies from different sources allows the most mature technologies, though still cutting-edge, to be identified and transferred into robotic behaviors. In turn, forming partnerships like this with universities and industry enables the universities and industry to improve the most useful technologies which can in turn be transferred back to military robots.

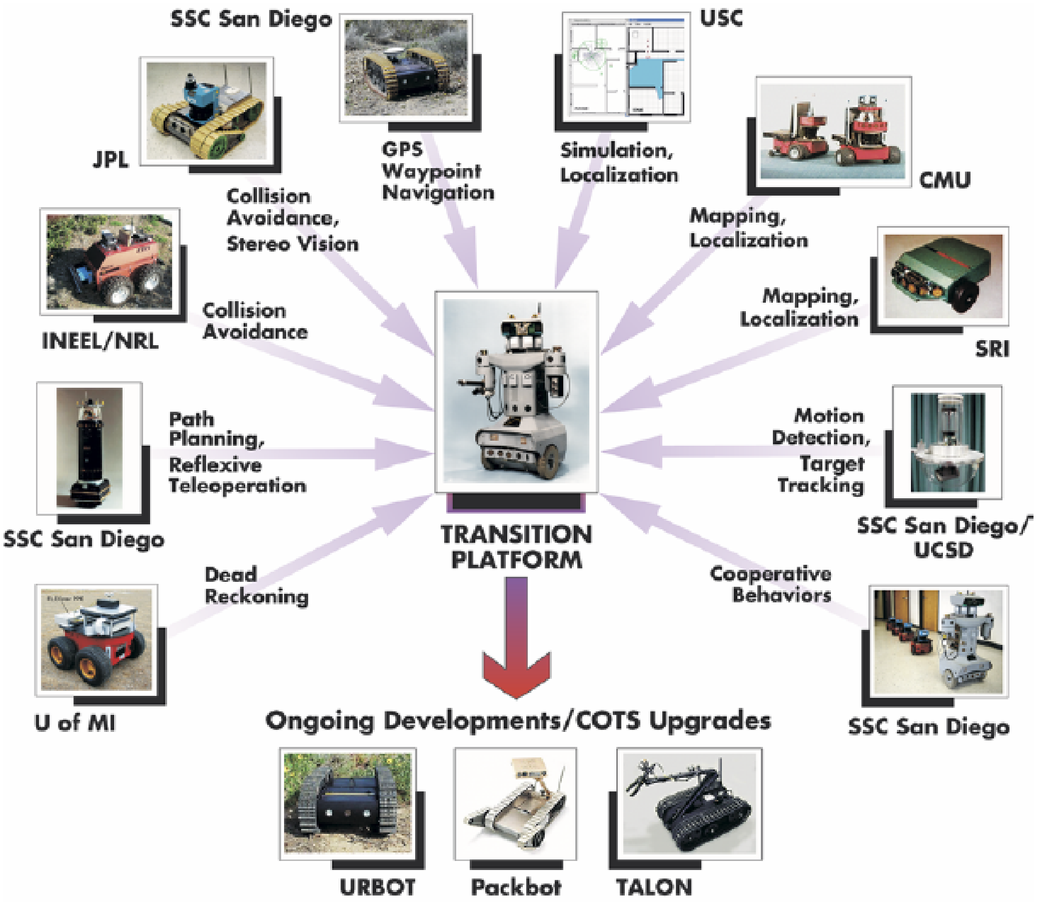

Figure 3.3: This shows an example of an earlier setup of the Technology Transfer Program at SSC Pacific (from 2004) [46].

Initially, transfered technology is tested on robots designated as research platforms. As technology matures, more behaviors are developed. Eventually, some of the behaviors reach a state where they are near-deployable. These behaviors are then transferred to platforms which are actually deployed, such as a *PackBot* or a *Talon* [30; 23]. Further testing and modification can then be applied to the deployable platform until it reaches a state where it is ready to actually be deployed in-theater. An example of an earlier setup of the Technology Transfer Program can be seen in Figure 3.3.

One particular behavior which the HPD methods in this thesis are developed to supplement is autonomous mapping capabilities using SLAM and a laser range finder [4]. This mapping behavior has reached a near-deployable status and is currently being further developed by the Unmanned Systems Branch at SSC Pacific [4]. Autonomous mapping has matured enough that it has transitioned from research robots to a deployable robot, specifically a *PackBot* which will be discussed in Chapter 5. The capability of autonomously constructing a map engenders good opportunities to apply other behaviors which would not have been practical without this mapping capability. The specific behavior explored in this thesis is human presence detection (HPD) which was chosen specifically because it would complement the autonomous mapping capability. For example, while a robot is constructing a map, it could be searching for humans and adding its detections to the map being constructed. Appropriate action could then be taken based on these detections. If a robot's mission, for example, were to rescue survivors from a disaster area, then constructing a map and detecting human survivors could help a rescue crew to act quickly while minimizing injury to themselves or to the survivors. If the robot's mission instead were to identify possible enemy threats in a bunker, then a map showing these threats would be valuable as well.

In part because of the Technology Program, there are many different behaviors on both the experimental robots and the deployable robots at SSC Pacific. The philosophy of this thesis is that robots should be viewed not as a set of disparate behaviors but as a single system. Therefore, if possible, these behaviors should be combined and/or executed in complementary fashions. This philosophy is discussed in the next Chapter.

Chapter 4

A Holistic Approach Towards Robot Autonomy

Rather than just focusing on improving a single aspect of robotic behavior, the other behaviors of the robot must be taken into consideration so they will not be compromised. In other words, the robot must be treated as an entire system. If an object detection algorithm is added to the robot, it must not interfere with the other tasks of the robot, such as its ability to move and avoid obstacles. By treating the robot as a system, an object detection algorithm might involve utilizing multiple sensory inputs or the robot's physical behaviors to improve the object detection algorithm or to help the algorithm fit into a niche to fulfill the robot's overall purpose.

4.1 A Purpose-Driven Life (for a robot)

In order for military man-portable ground robots to be used and to increasingly participate in more dangerous missions, thereby decreasing the involvement of human soldiers in the more dangerous tasks, the robots must naturally be more of an asset than a nuisance. If a particular technology does not add a benefit to the robot which will justify its existence, then that technology should not be used until it has matured. By considering the robot's purpose and viewing it as a system, the maturation of a particular technology might be fulfilled sooner than it would if it were considered apart from the robot.

For example, if a HPD method existed that correctly detected humans 100% of the time while inside of a bunker but produced false alarms 20% of the time (i.e. falsely classified background clutter as human), then the HPD method might be considered mature if it were only used for bunker exploration in certain situations and the detection were added to a map for further verification. This would make the HPD method much more likely to be an asset to the soldier than a nuisance if put in this situation. On the other hand, if the robot's task were to shoot enemies once the HPD method detected a human, then the robot would probably expend all of its ammunition by shooting at 20% of everything it sees and destroy the building in the process. This situation would make it a nuisance to soldiers who could do a much better job clearing out a building themselves.

In terms of efficiency, consider a hypothetical robot which has a task of mapping a bunker. The robot is capable of mapping a ten room bunker in ten minutes (about one minute per room). If it used a HPD method that detected people 100% of the time with no false alarms but took five minutes to handle each detection, then this would invalidate the robot's primary purpose of mapping the bunker by causing its exploration time to jump from ten minutes to possibly several hours depending on how often it checked for humans. In this case, it might be better to settle for a HPD method that runs in real-time but only detects people 80% of the time.

The required detection rate would depend on the *purpose* of the robot mapping the bunker. If the robot's purpose was to explore a disaster area for survivors, then the robot would probably be required to detect people 100% of the time with the false-positive rate being less significant if detections were added to a map for human verification. On the other hand, if the robot's purpose was to follow a group of friendly soldiers through a bunker and to create a map as they went, then having a lower human detection rate would probably be acceptable but having a high false-positive rate would not, as it would be better for the robot to lag behind a bit than to end up travelling toward its false-positive classifications and away from the soldiers. If, for some reason, the robot's purpose was to stand guard in a hallway and alert friendly soldiers if any enemy soldiers appeared using an audible signal, then the robot would probably require both a high human detection rate and a false-positive rate of 0% – especially if the robot were armed.

4.2 Avoiding Sensor Overload

Computer vision on a robot should be analogous to eyesight in humans. When human soldiers explore a bunker on a mission, they do not stop to analyze everything they see. Instead, they notice objects of interest. If they notice a table or a chair, they may only notice it enough to avoid bumping into it unless they are specifically looking for tables and chairs. Humans naturally avoid sensory-overload by effectively narrowing the scope of the search problem. This can be seen in the example of finding lost keys which Kogut, et al mention in their paper:

> When humans enter an area to search for some particular item, such as lost keys, we generally do not look at every possible location within that area and mentally compare it to every object we have previously seen. To determine if our lost keys are present, we might search for distinguishing features, such as the shiny key ring or the serrated edges of keys. Furthermore, we generally don't wander outside into our neighbor's yard looking for our keys if there is no reason to suspect they were lost there. When moving through areas where our lost keys are not likely to be, we more likely concentrate on detecting and recognizing objects that could cause us to trip and fall. We generally confine our searches to likely places, such as a table in the kitchen, and do not consciously think about all of the other objects that we have previously observed.[27]

In terms of the robot, this translates to utilizing sensors or behaviors which facilitate the detection of certain abstract features of objects very well. For example, the iRobot *PackBot* 500 with a mapping kit has a laser range finder (LIDAR) which is very good at detecting solid objects at its level [31]. If an image-based HPD algorithm only triggered after the LIDAR detected a solid object, then the situation where, for instance, a human is standing up would be much more likely to occur in these situations so the image-based HPD algorithm could afford to have a higher false-alarm rate than it could if the LIDAR were not present. Kogut ,et al, specifically, combine the LIDAR with an image-based HPD method in their paper for the purpose of following a person [38].

4.3 Approximating Biological Behavior

Robots are currently not nearly as good as humans at many tasks. In order to make robots capable of performing tasks that humans would perform, they must compensate for their weaknesses by utilizing other sensors and exploiting the situation they are in. When this is not adequate, robots must rely on humans to compensate for their weaknesses.

A compromise between a tele-operated operator control unit (the human controls everything) and a fully-autonomous robot is the Multi-Robot Operator Control Unit (MOCU) developed at SSC Pacific by Bruch, et al [47]. This interface is designed to monitor multiple robots at once (while they perform autonomous behaviors) and to send commands to them, much like orders are sent to human soldiers. At any given time, a robot being monitored can be controlled manually or sent ”orders” such as a command to follow a specific path, to navigate waypoints, or to otherwise travel to a specified location. In contrast to the tele-operated OCU requiring the human soldier to pay complete attention to moving the robot, the MOCU interface allows the human soldier to suggest specific pathways while the robot is free to figure out how to navigate them – effectively allowing a single human soldier to command multiple robots rather than the current situation in-theater where multiple human soldiers are required to command a single robot.

Another lower-level approach to commanding a robot is where the robot serves as sort of a sidekick to the soldier, much like a dog. This concept was initially outlined in the *Warfighter's Associate* concept introduced by Everett, et al [19]. A demonstration of the *Warfighter's Associate* concept is the *BirdDog* human-computer-interaction sys-

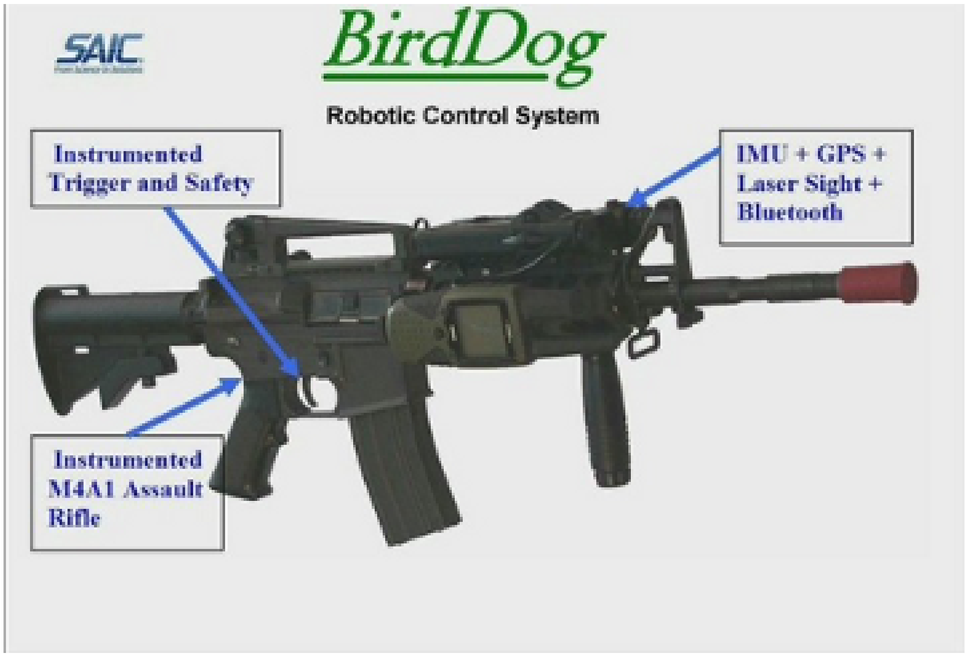
SAIC
BirdDog
Robotic Control System
Instrumented Trigger and Safety
IMU + GPS + Laser Sight + Bluetooth
Instrumented M4A1 Assault Rifle

Figure 4.1: This image, taken from the paper by Fellars, et al, shows a diagram of the Birddog Warfighter Sensor System [21].

tem, developed by SAIC of Englewood, Colorado and described by Fellars, et al [21]. BirdDog involves a sensor fitted to an M4A1 rifle that communicates the status of the weapon to a PDA, as shown in Figure 4.1. The PDA then transmits the weapon's status, the direction the weapon is being pointed, and the GPS location of the warfighter [21]. The robot can then use this data to respond to the perceived actions of the warfighter [21]. With the *BirdDog* concept, information that might otherwise be very difficult for the robot to obtain but very easy for the human soldier to obtain can be communicated to the robot without distracting the soldier. This fits with the idea that the robot should be an asset and not a nuisance.

These methods of commanding the robot enable the robot to operate with technologies that might otherwise not be mature enough for deployment. They strike a balance between allowing the robot full autonomy and fully tele-operating the robot, enabling it to be more of an asset to the human soldier. When required technologies mature enough for deployment, the robot can be gradually transitioned into a more important role in-theater rather than being forced to stay back in the lab to await future developments.

Chapter 5

Target Robotic Platform For This Thesis

5.1 Hardware

The target platform for this thesis is the same robot used by Ahuja, et al in their paper [4]. This robot has an iRobot *PackBot* Scout chassis with a first-generation iRobot Navigator Payload [4]. It is very similar to the *PackBot* 500 with the mapping kit featured on iRobot's website, which is shown in Figure 5.1 [31]. This platform is not yet deployment-ready because the mechanical portions of the mounted 360-degree laser range finder (LIDAR) make it relatively fragile. For deployment, the LIDAR could easily be replaced with a solid-state or otherwise ruggedized LIDAR when one becomes available. Because of the Technology Transfer Program, as mentioned in Chapter 3, this robot is constantly receiving improved technology as it becomes available. For instance, in future tests, the existing LIDAR will be replaced with a smaller and lighter one which is shown in Figure 5.2. The LIDAR is mostly used to generate maps though its data could be used to help with object detection tasks, such as by detecting the presence of obstacles with a SLAM-generated map which might include specific objects to detect [4]. The *PackBot* also comes with a wide-angle color ”drive-camera” which is fixed in

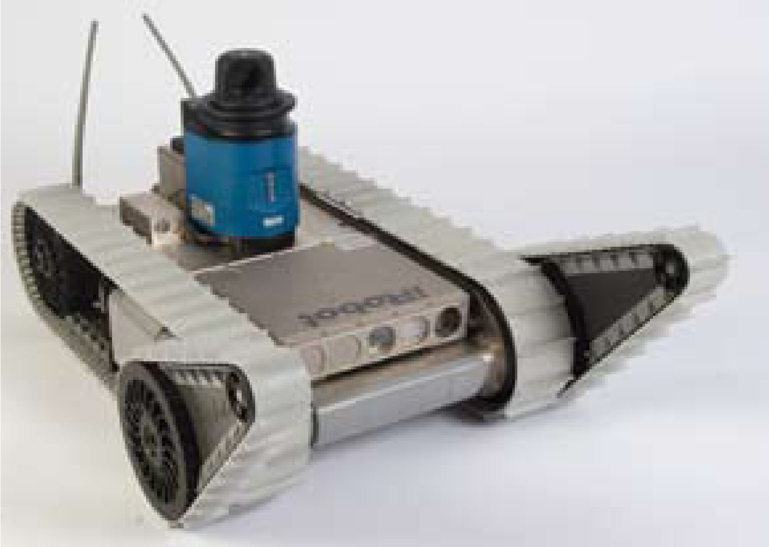

Figure 5.1: This image, taken from the iRobot website, shows the iRobot *PackBot* 500 with the mapping kit [32].

the front [31].

The *PackBot* has been modified by the Unmanned Systems Branch at SSC Pacific to use a FLIR Photon 320 far-infrared camera, similar to the one shown in Figure 5.3 [56; 22]. The data from this camera is converted to grayscale images with a resolution of 320x240 pixels. Some FLIR cameras require cryogenic cooling but the Photon 320 does not because it uses a microbolometer array as its detector which uses infrared-absorbing material instead, as shown in Figure 5.4 [22; 42; 37].

5.2 Software

5.2.1 Autonomous Capabilities Suite

The software used on the robot to facilitate autonomous behaviors is the Autonomous Capabilities Suite (ACS) developed by SSC Pacific [4]. The programming language for ACS is mostly C++. From the paper by Ahuja, et al, ”ACS is a modular software architecture supporting development and maturation of new payloads, devices,

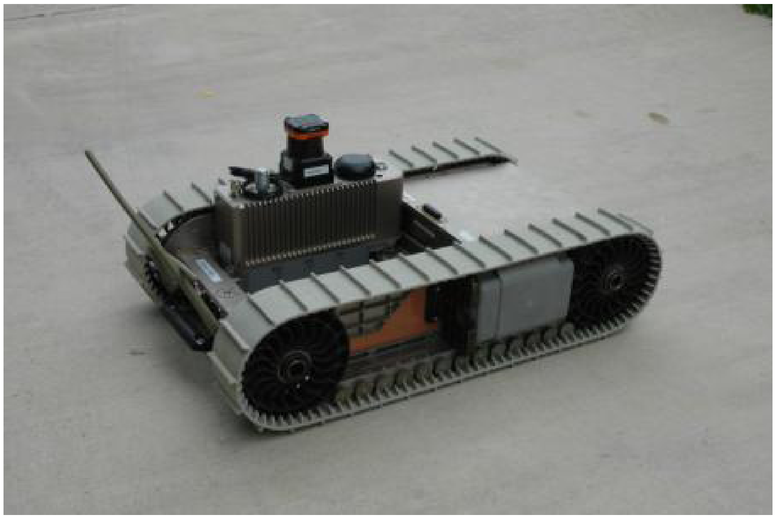

Figure 5.2: This image, taken from the paper by Ahuja, et al, shows a *PackBot* with a next-generation LIDAR mounted on it. [4].

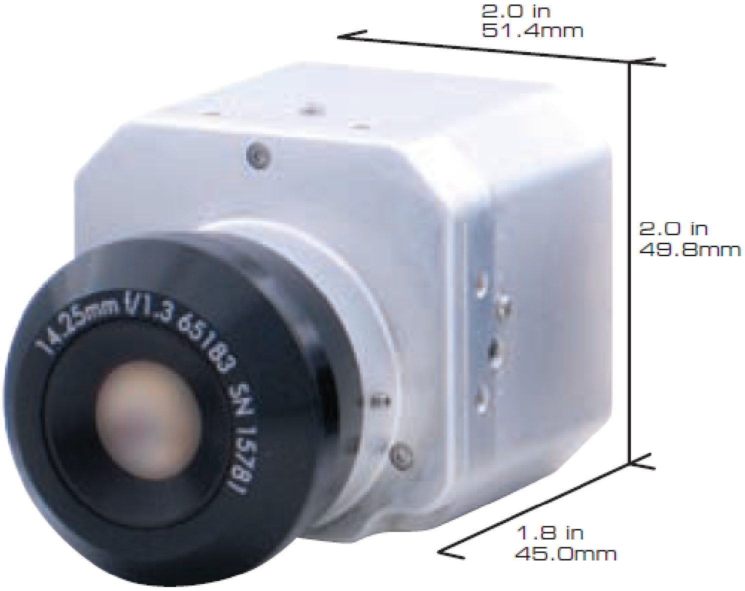

Figure 5.3: This image, taken from the FLIR Systems website, shows an example of a Photon far-infrared camera [22].

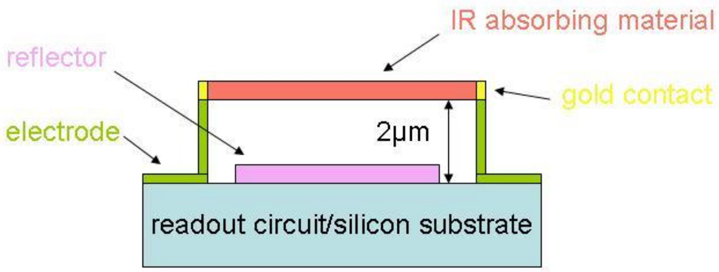

Figure 5.4: This image, taken from Wikipedia, shows a cross-sectional view of a single pixel from a microbolometer array. This is the driving technology behind portable, uncooled high-resolution infrared cameras [37].

perceptions, behaviors, human-robot interaction techniques, and communication protocols on unmanned systems." It was designed to treat the robot as a system composed of behaviors, drive functions, and sensor-handling routines. It is multi-threaded so that while the robot is driving around, it can simultaneously avoid obstacles, create a map, and detect anomalies. When the HPD algorithm presented in this thesis reaches maturity in its development process, it can easily be integrated with the ACS framework. This software suite is similar to the Player-Stage open-source software in that it is designed to handle the robot as a unified system rather than as a disjointed set of capabilities, which is consistent with the overall philosophy advocated by this thesis [26]. This software fits in well with the Technology Transfer Program in that when a new sensor or behavior becomes available, it can be added to the existing software framework as a modular component without compromising other behaviors or requiring new software to be written from scratch. A chart showing the organization of the ACS software framework can be seen in Figure 5.5.

5.2.2 Open Computer Vision Library

The Open Computer Vision Library (OpenCV) is the primary open-source computer vision software used for image processing in this thesis [53]. OpenCV is very popular and was used by the winning team (Stanford) in the 2005 Darpa Grand Chal-

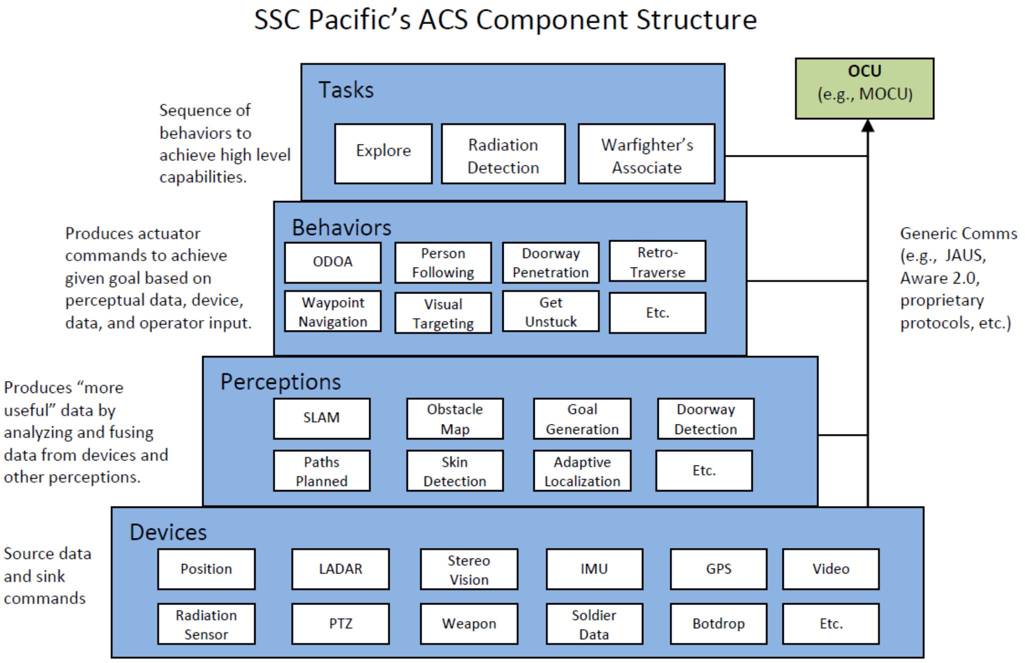
SSC Pacific's ACS Component Structure
Tasks
Explore
Radiation Detection
Warfighter's Associate
Sequence of behaviors to achieve high level capabilities.
Behaviors
ODOA
Person Following
Doorway Penetration
Retro-Traverse
Waypoint Navigation
Visual Targeting
Get Unstuck
Etc.
Produces actuator commands to achieve given goal based on perceptual data, device, data, and operator input.
Perceptions
SLAM
Obstacle Map
Goal Generation
Doorway Detection
Paths Planned
Skin Detection
Adaptive Localization
Etc.
Produces "more useful" data by analyzing and fusing data from devices and other perceptions.
Devices
Position
LADAR
Stereo Vision
IMU
GPS
Video
Radiation Sensor
PTZ
Weapon
Soldier Data
Botdrop
Etc.
Source data and sink commands
OCU
(e.g., MOCU)
Generic Comms (e.g., JAUS, Aware 2.0, proprietary protocols, etc.)

Figure 5.5: This image, taken from the paper by Ahuja, et al, shows the structure of the ACS software framework. It is designed to be cross-platform so it can run on multiple robots and it achieves this via its modularity and messaging capabilities. [4].

lenge to detect desert roads by Dahlkamp, et al [11; 61]. OpenCV includes function headers for the C programming language so the HPD algorithms in this thesis can be readily integrated with the ACS framework for further development or testing.

5.3 Behaviors

As a result of the ACS software, the *PackBot* has many behaviors. The three behaviors most specifically relevant to this thesis are [4]:

1. its ability to generate maps using the Simultaneous Localization and Mapping algorithm (SLAM)
2. the ability to plan paths
3. the ability to explore its environment

The ”explore” behavior allows the robot to explore an area on its own. While it is exploring, the SLAM algorithm allows the robot to avoid obstacles, keep track of its current position, and generate a map of the area it is exploring using its LIDAR. The robot can go to specific places on the map using its path-planning capabilities. While it is performing these two behaviors, it uses SLAM to avoid obstacles and to generate a map of where it has been.

The combination of these three behaviors allows the robot to perform autonomous mapping which, even in the absence of other autonomous capabilities, is useful enough to justify deploying the robot on its own and help soldiers to carry out their missions safely. These behaviors also provide a good foundation for integrating object detection capabilities with the robot’s behaviors. While the robot is exploring, for example, an HPD algorithm could be running in the background to detect what might be humans. These detections could then be added to a map for further verification by a human soldier using the generated map. The robot could also modify its behavior to engage the human, avoid the human, or position itself for a better look at what it thinks is a human.

Chapter 6

Visual Human Presence Detection (VHPD)

There are two separate parts to the HPD method used in this thesis. This portion consists of detecting human-like shapes from infrared images. This will hereinafter be referred to as visual HPD or VHPD for short. The second portion consists of simulating a modification of the robot's motion to see if physically moving the robot in response to potential human detections would cause the HPD algorithm to achieve better detection results. This will hereinafter be referred to as HPD with physical repositioning or HPD-PR for short. This chapter discusses the VHPD method while the next chapter discusses the HPD-PR method.

6.1 Training Set

6.1.1 Single-Scale Positive Training Set

A set of positive training images were acquired from a Photon 320 infrared camera, which was described in Chapter 5. Even though the target platform is a *PackBot*, the images were acquired from a camera (the same model of infrared camera) mounted on an iRobot *All Terrain Robotic Vehicle (ATRV)* research robot. The *ATRV* is one of

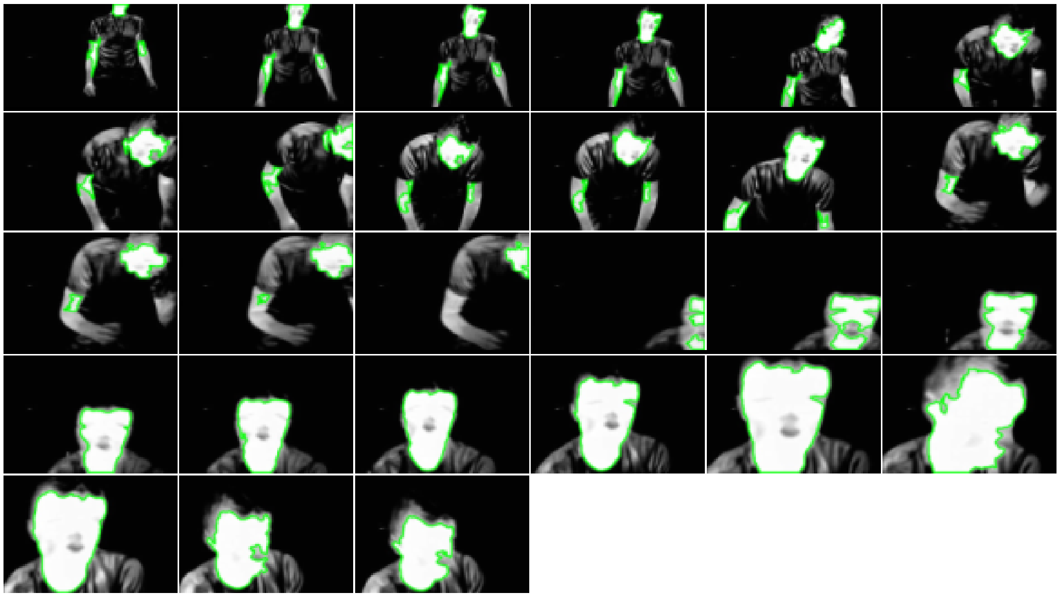

Figure 6.1: Here are some of the positive training examples of a single person. The green outlines represent contours extracted after thresholding the image (see Section 6.2.2).

the transition platforms used by the Technology Transfer Program at SSC Pacific to test behaviors and sensors before they are transferred to a *PackBot* [46]. The main difference between this robot and the *PackBot* with respect to the camera is that the camera is mounted about a foot higher on the ATRV than it is on the *PackBot*.

The initial positive training images consist of 567 images of a single person, in a cool indoor environment, at a close yet variable distance in different positions with respect to the camera (e.g. some without the head showing, some with partial occlusions, etc.). This was done in contrast to the work by Kogut, et al where the largest region (usually the head of a person) was classified as human [38]. A small subset of these images, which captures most of the variation within the training set, is shown in Figure 6.1. The infrared camera has built-in automatic contrast adjustment which was disabled as was done by Kogut, et al [38]. The infrared camera also includes on-board thresholding which was adjusted so that human skin stands out the most and much of the background is filtered out.

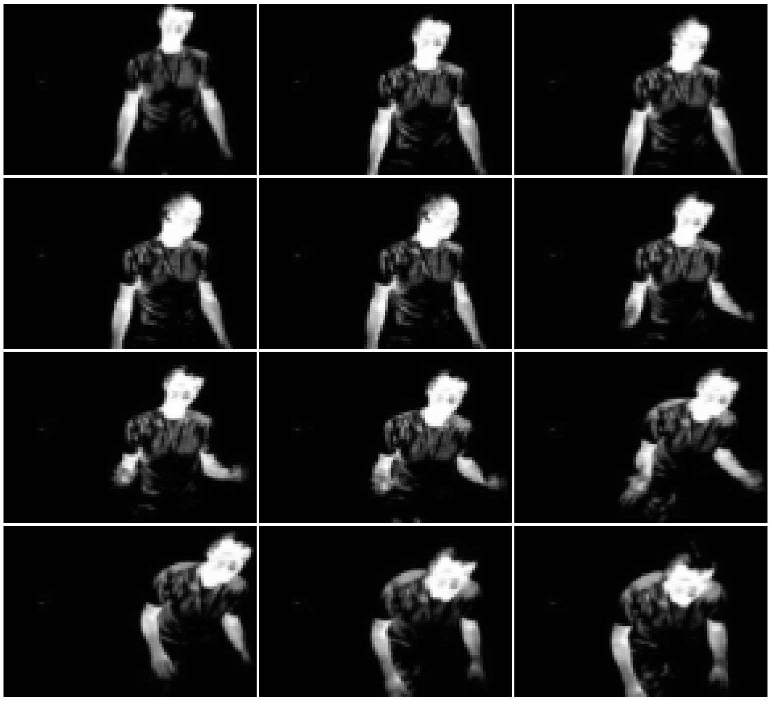

Figure 6.2: Here are some of the positive training examples used for the multi-scale positive training set.

6.1.2 Multi-Scale Positive Training Set

The initial positive training set was constructed with a moderate amount of variation. This was done so that various feature types could be tested for their ability to separate the positive and negative training data, of which the polar histograms were found to perform the best. For subsequent testing and experimentation, the initial training set was manually pruned to include conditions that the robot would be more likely to encounter in order to narrow the scope of the problem and hopefully yield more practical classifiers that would be better at detecting the more common cases.

To prune the training set, some positive training images which exhibit partial occlusion of the head and body were initially removed. Extremely close-up images were removed as well. Some examples of these removed images can be seen in Figure 6.3. It was reasoned that if a robot were exploring a bunker and searching for humans, it would be much more likely to encounter humans from a distance as opposed to right next to

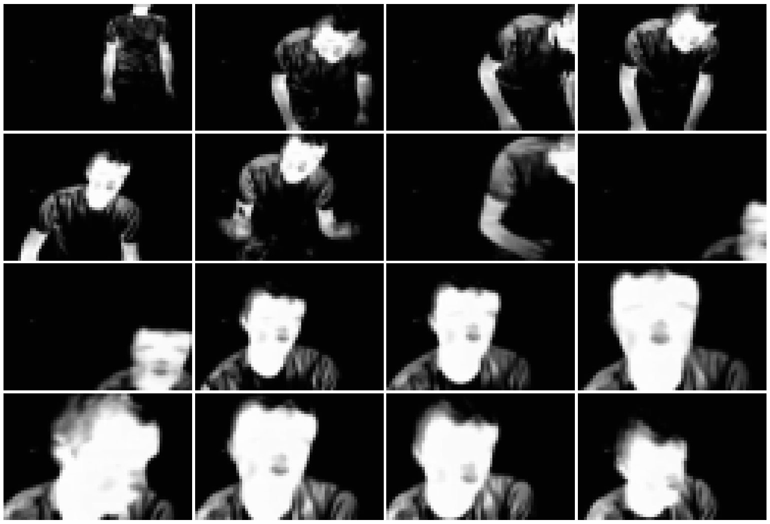

Figure 6.3: Here are some of the positive training examples that were excluded when creating the multi-scale positive training set.

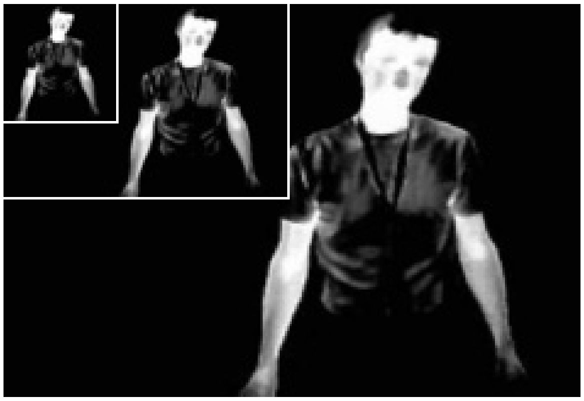

Figure 6.4: Here is an example of the three different categories of image sizes in the multi-scale positive training set. The image sizes are 70x48, 116x79, and 236x161 for the small, medium, and large image scale categories, respectively.

the camera. It was also reasoned that contours from far away might have very different shapes from contours close-up, making it difficult to match one to the other. To facilitate tests of this theory, the remaining training images were scaled to simulate a person at three different distances, as shown in Figure 6.4. The negative training examples were not modified as they already include images of buildings from a military training facility taken at different distances [4].

There are three different size categories for the training set: small, medium, and large. Each size category consists of 126 positive training images taken from the single-scale positive training set. The original positive training images were resized by three different percentages: 20%, 33%, and 67% for the small, medium, and large image categories, respectively. These percentages were determined by looking at a set of images of a person walking towards the infrared camera from the end of a long hallway. The images were then scaled until the person in the images was close to the size of the person in these hallway images near the beginning, middle, and end of the walk. This yielded image resolutions of 70x48, 116x79, and 236x161 for the small,medium, and large image categories, respectively. An example of the relative image sizes can be seen in Figure 6.4. Some of the positive training images from this set are shown in Figure 6.2.

6.1.3 Negative Training Set

The negative training images consist of 1320 images of buildings in a hot, outdoor environment at a MOUT (Military Operations on Urban Terrain) test site [34]. This test site is described by Ahuja, et al in their paper [4]:

> The range consists of residential sections, including a gas station, houses, apartment buildings, a school, a playground, a business district consisting of a hotel, office buildings, and a town square. There are a total of 29 buildings (one, two- and three-story), 14 that are intact and 15 that have been partially damaged. In addition, there are nine ghost buildings to represent buildings that have been completely destroyed. Testing in FY08 was scheduled every month and has been solely on paved roads and in cluttered, single story buildings [4].

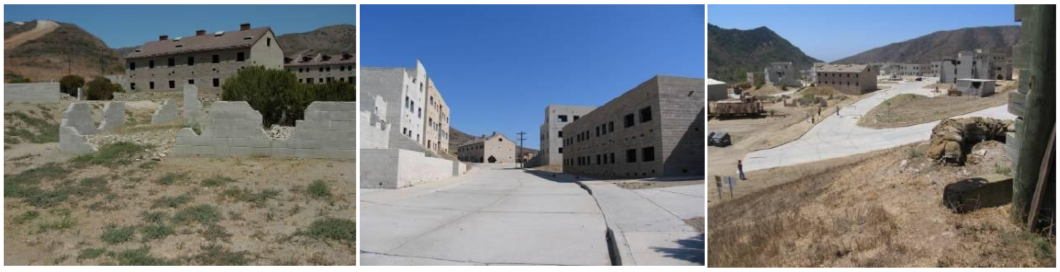

Figure 6.5: This figure shows three color images of the MOUT test site, taken from the paper by Ahuja, et al [4]. This is where the negative training images were captured for this thesis.

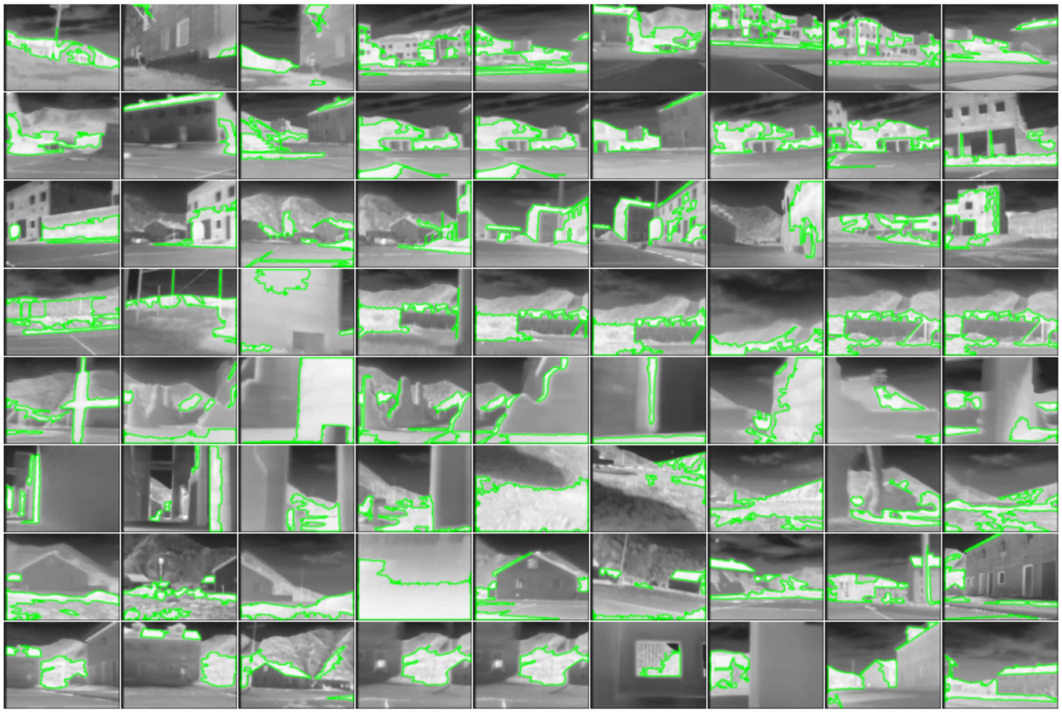

Figure 6.6: Here are some of the negative training examples of buildings at the MOUT test site [34]. The green outlines represent contours extracted after thresholding the image (see Section 6.2.2). Notice how the brightest regions of the images tend to be where the contours are extracted.

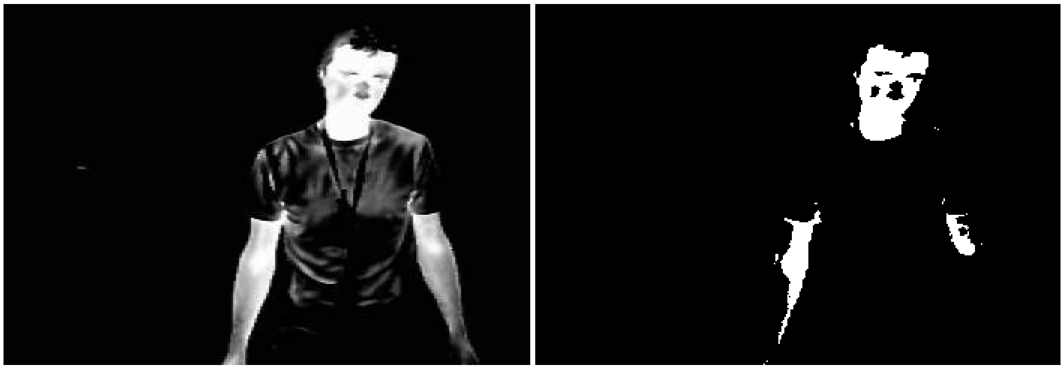

Figure 6.7: Here is an example of the initial thresholding of the thermal images with 210 being the threshold value.

Color photos of the test site can be seen in Figure 6.5. The negative training images were captured using the same infrared camera as the positive images with the difference being that the camera for the negative images was mounted on the *PackBot* rather than the *ATRV*, resulting in only a slight difference in the height of the camera. Like the positive training images, a small subset of the negative training images, which captures most of the variation within the training set, is shown in Figure 6.6.

6.2 Extracting Features

6.2.1 Thresholding For Temperature

At the core of the VHPD algorithm, shape is used to detect humans. In order to extract shapes that might be human, the grayscale images from the infrared camera are thresholded using OpenCV to generate binary images [53]. Anything in the images with a pixel value greater than 210 (with 255 being pure white and 0 being pure black) is set to 255 and anything less than or equal to 210 is set to 0. The threshold value of 210 is the same value used by Kogut, et al in their paper [38]. An example of this is shown in Figure 6.7. This thresholding operation causes many potential false-positive classifications to be eliminated right away. On the other hand, if the environment is warm, objects in the environment tend to show brightly which causes humans to blend into the background.

6.2.2 Finding Contours

Once the image has been thresholded and binarized, contours are found for the remaining white regions of the image. The contours are computed using the OpenCV function *cvFindContours()* with a simple chain approximation of the resulting polygons such that only the endpoints of horizontal and vertical sections of the polygons are stored. This is the same method employed by Kogut, et al in their paper [38]. Examples of these contours can be seen as green outlines in Figures 6.1 and 6.6.

6.2.3 Filtering Out Noise

When the image is thresholded, very small white regions often appear throughout the image (effectively, noise). To remove these regions, all contours with a pixel area of less than a threshold value are thrown out. This presents a problem when people are to be detected from far away. If the area threshold is too large, then people far away won't be detectable. If the area threshold is too small, then noise results in contour regions which are possibly too small to have shape characteristics that are distinguishable between human and non-human classifications.

6.2.4 Classifying Based on Contour Pixel Area

Different sized contours are sent to be classified by one of three separate classifiers trained to handle contours of a particular size category. These three size categories correspond to the three size categories of the multi-scale training set of small, medium, and large image scales. To determine which size category each contour fits into, each contour is checked to see if its pixel area falls within one of three ranges of threshold values, each represented as $(\mu, \tau]$ where μ is the lower bound and τ is the upper bound for the pixel area. Contours that are less-than or equal-to μ for the small size category or greater than τ for the large category are skipped. The specific values used for μ and τ for each size category will be explained in the experimental sections of this thesis.

For humans that are too far away to produce any sort of distinguishable contour, A possible solution is to use some sort of filter-based classifier, such as the Haar-like features used by Viola and Jones for humans that are far away [70].

6.2.5 Constructing Feature Vectors

Once contours are found and the smaller ones are discarded, feature vectors are constructed from the remaining contours. In early experiments, several different types of feature vectors were constructed and polar histograms were found to yield the lowest error rates so they will be described.

Polar Histograms

To construct polar histograms, the contour points are first centered by subtracting the centroid – causing each contour point to be remapped to a local coordinate system relative to an origin at the center of the contour. The Cartesian coordinates of each contour are then converted to polar coordinates, as shown in Equation 6.1, yielding an angle θ and distance ρ for each point (x, y) with respect to the origin (i.e. the centroid, which is now zero since the coordinates have all been centered).

$$\begin{aligned} \theta &= \arctan(\frac{y}{x}) \\ \rho &= \sqrt{x^2 + y^2} \end{aligned}$$

Once the polar coordinates are calculated, two separate histograms are constructed from the points – one for the angles and one for the magnitudes. The lower and upper bounds for the angle histogram are set to $0°$ to $360°$, respectively. To compute the lower and upper bounds for the magnitude histogram, a bounding rectangle is first fit around the contour. The lower bound for the magnitude histogram is then set to 0 and the upper bound is set to the distance from the center of the bounding rectangle to one of the corners. The histograms are initialized to have 25 bins each, as this was found to yield the lowest error rate.

6.3 Classifying Features

Feature vectors are constructed from the training data and stored. New feature vectors are classified by finding the K-nearest-neighbors [10].

6.3.1 K-Nearest-Neighbors

The K-nearest-neighbors algorithm (K-NN) works by taking a new feature vector and finding the K closest neighbors using some sort of distance metric. For example, if the value of K is chosen to be 10, then the 10 closest (i.e. the smallest value from the distance metric) feature vectors from the training set are used to determine the final classification. Usually, out of the K closest neighbors, the class label of each of the neighbors is tallied up and which ever class has the most votes is what the feature vector in question is classified as. For example, if K is 10 and 6 of the 10 closest neighbors are from the positive training set, then the feature vector in question is classified as positive (i.e. human).

In this thesis, a value of K=1 was found to produce the lowest error rates so it was used for all the experiments. This also simplified the experiments so that the results of using three separate classifiers for each of the multi-scale training categories, respectively, could be better analyzed.

Chi-Squared Histogram Matching

The distance metric used for matching the feature vectors is a weighted sum of Chi-Squared distances between polar histograms. The Chi-Squared metric is shown in Equation 6.1 where $h_i(n)$ and $h_j(n)$ represent normalized histograms [50]. The weighted sum distance metric is shown in Equation 6.2 where $h_{\theta i}$ and $h_{\theta j}$ are the polar angle histograms from two different feature vectors and $h_{\rho i}$ and $h_{\rho j}$ are the polar magnitude histograms.

$$\chi^2(i,j) = \frac{1}{2}\sum_{n=1}^{N} \frac{[h_i(n) - h_j(n)]^2}{h_i(n) + h_j(n)} \tag{6.1}$$

$$distance = \frac{1}{3}\chi^2(\rho i, \rho j) + \frac{2}{3}\chi^2(\theta i, \theta j) \tag{6.2}$$

The Chi-Square distance represents the amount of similarity between histograms. The result of computing the Chi-Squared metric between histograms is a value between 0 and 1 where 0 represents a strong similarity and 1 represents a weak similarity. The weights in Equation 6.2 were arbitrarily chosen because they were found to yield the lowest error rate.

Chapter 7

Human Presence Detection with Physical Repositioning (HPD-PR)

7.1 Detection Phases

HPD-PR is based on the notion that humans which would otherwise be too far away to detect reliably can be detected if the robot is able to spot them from a distance and then move closer to verify that what it sees is in fact a human. This concept is explored experimentally in this thesis although it has many opportunities for improvement.

In order to determine how far away a potential human is, the area of each contour is considered. This is based on the idea that as the camera gets closer to a human, the area of the contours representing the human tend to increase. These larger contours contain more detailed information on their shape than the smaller contours which might translate to more useful information that would make them easier to detect. The steps to detecting a human with HPD-PR might proceed as follows:

1. Initial Detection: The VHPD method is run.

2. Motion Compensation: If a contour is detected as positive but it has a small or medium-sized area, then move towards that contour.

3. Final Detection: Step 1 is repeated.

This treats the three different classifiers in the VHPD method as three different layers of likelihood that a particular region belongs to a human. If a contour has an area that falls into the small or medium categories and is classified as human, then the robot has a reason to move closer so that the contour falls into the large category. As the contour becomes larger, it is hoped that this will increase the probability that it will be correctly classified as human or non-human.

Although the VHPD method tested in this thesis uses K-NN with K=1 for each of the three different classifier scales, classifiers with larger values of K or other types of classifiers which allow a threshold to be adjusted could be used in their place. The classifiers for the small and medium contour areas could be adjusted so that they have higher false-alarm rates but higher true-positive detection rates while the classifier for the large contour area could be adjusted to have lower false-alarm rates. This would roughly translate to the robot pursuing regions which it thinks have a higher likelihood of being human and ignoring regions which it thinks have a lower likelihood of being human.

7.2 Reacting To Recognition

There are several primary actions that the robot could take upon detecting objects of interest, such as humans. These would include target acquisition, avoidance, and alerting human soldiers (e.g. via modifying a map, producing an auditory signal, or alerting a human through a PDA). When a robot exploring a bunker detects objects of interest, its capability should remain useful despite false classifications. By adding objects of interest to a map of the bunker, a human can verify that the added objects are what the robot classified them as and also use these objects to determine how the bunker is used. These concepts are not performed in this thesis but are critically important future steps to take in order to make sure that the robot fulfills a practical role.

7.3 Testing Images

In order to simulate the HPD-PR method, nine different image sets were acquired from the infrared camera of an iRobot ATRV Sr. robot. These images show a human walking towards the camera from a significant distance away. The complexity of these images varies from having a sparse, cool background to having a cluttered, hot outdoor background. The outdoor temperature was around $60°F$ though it was during the afternoon so the sun had an opportunity to heat up various surfaces. An example of three images from each of the nine sets can be seen in Figures 7.1 and 7.2. Specifically, each set of images shows the following:

1. A human indoors walking up a hallway
2. The same as item 1 except with a different human
3. The same as item 1 except with a different human
4. A human outdoors walking towards the robot in front of a garage on a hot driveway
5. The same as item 4 except while holding a hot laptop
6. The same as item 4 except while holding a hot laptop and not facing the robot
7. The same as item 4 except with the human at a higher temperature
8. A human indoors walking up a hallway away from the robot
9. A human outdoors walking away from the robot in a hot parking lot

By showing a person at different scales, simulating the robot moving towards a person, it can be seen whether or not at some particular point in the series the robot would achieve an initial detection and therefore have reason to move closer to the suspected human. At the same time, the accuracy of the classifiers used can be tested and it can be determined whether or not better detections are achieved in practice as the human moves closer to the camera.

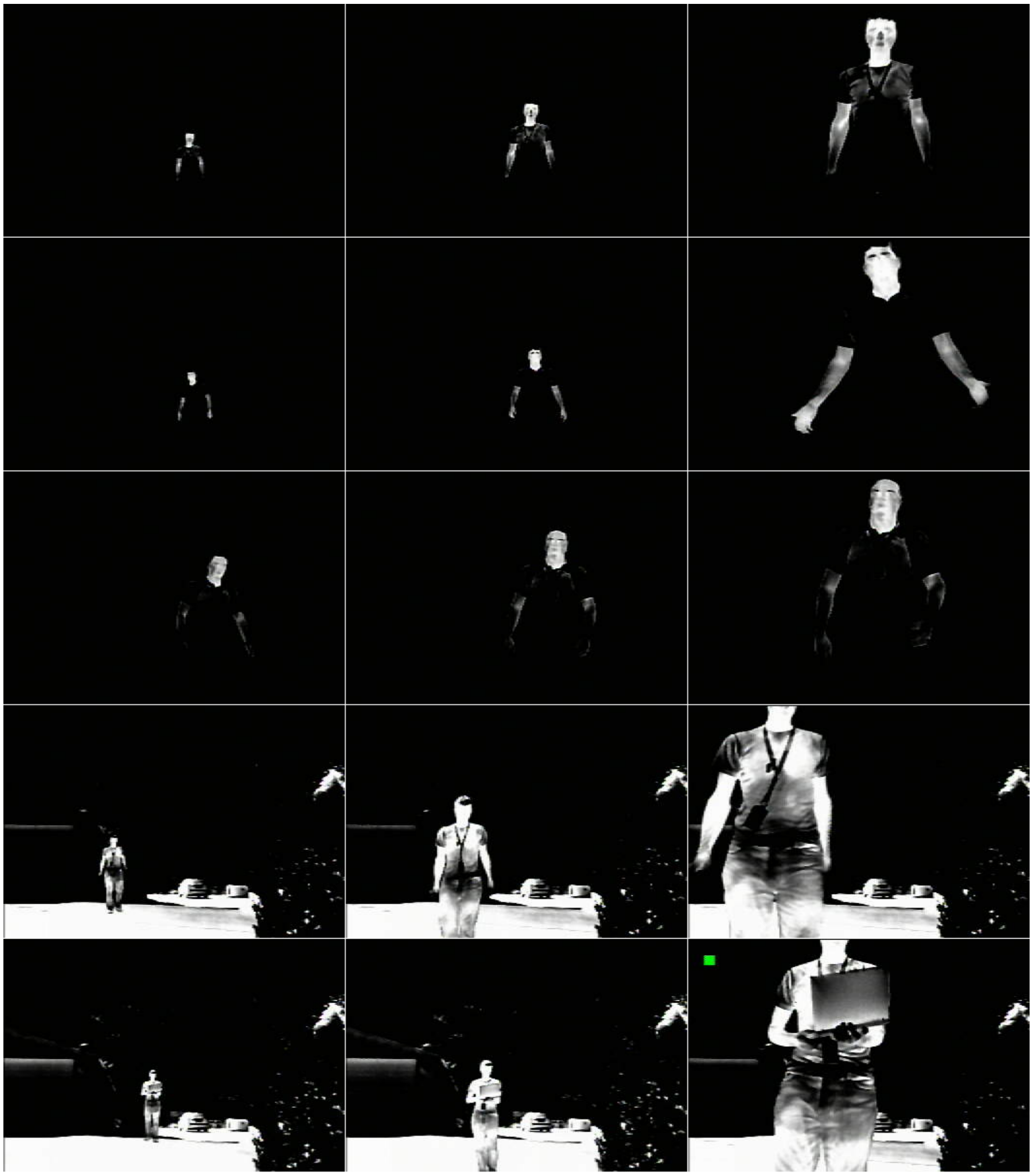

Figure 7.1: This figure shows an example of three different-scaled images from each of the first five sets of testing images. The image sets for the first three rows were taken indoors and the remaining two were taken outdoors.

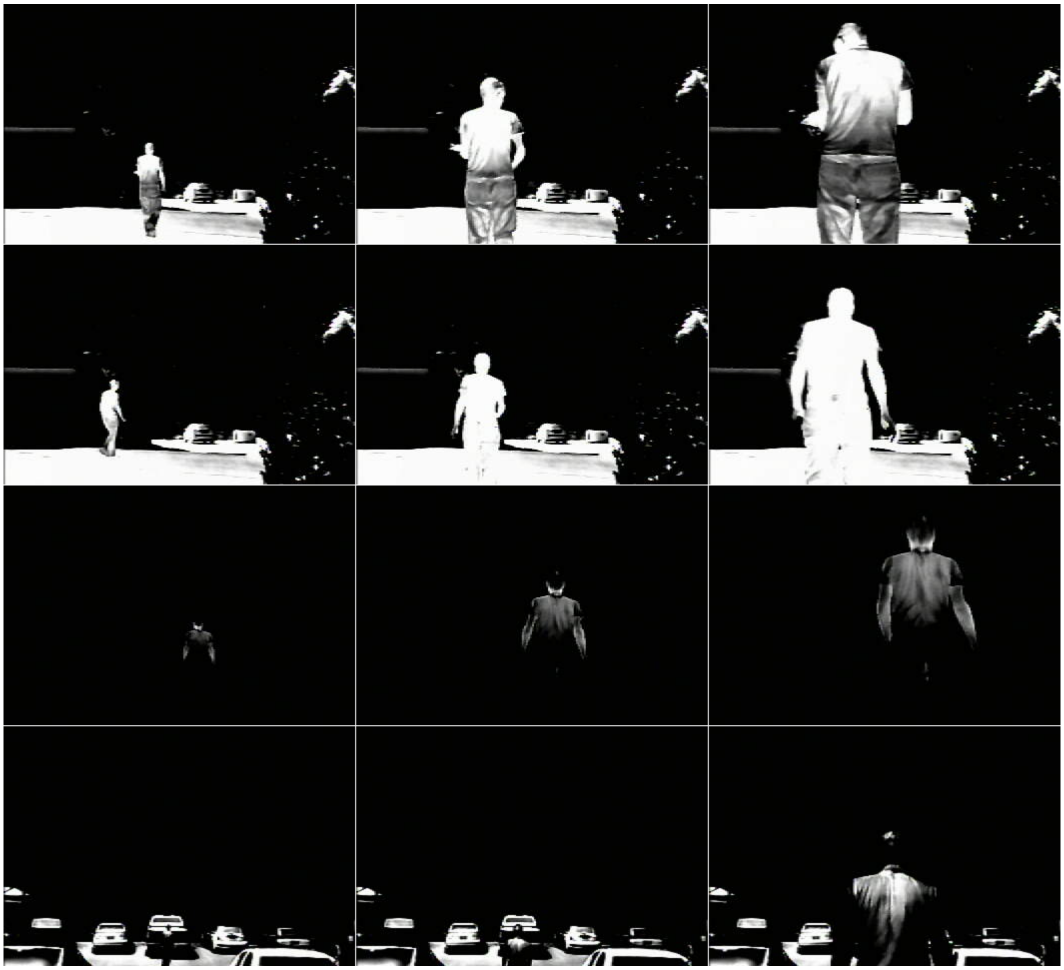

Figure 7.2: This figure shows an example of three different-scaled images from each of the last four sets of testing images. All rows except for the third show images taken outdoors.

Chapter 8

Experiments and Results

As mentioned previously, the robot's motion is simulated in order to produce solid experimental results. When the VHPD algorithm matures enough, then it will be transferred to a robot for further testing. When one of these robots explores an area, it is very difficult to get the robot to follow the same path twice due to all the logic being executed, such as how it explores an area and how slight variations in the amount of friction between its wheels or tracks and the ground causes the robot to follow slightly different paths. In order to simulate the motion of the robot in this thesis, the humans to detect move closer to the robot to simulate the robot moving closer to them upon an initial detection.

8.1 Performance Metrics

8.1.1 Leave-One-Out Cross-Validation (LOOCV)

To evaluate the performance of the VHPD classifier, Leave-One-Out Cross-Validation (LOOCV) is used [33]. LOOCV involves removing a single feature vector from the training set and classifying it using the remaining feature vectors. The removed feature vector is then put back into the training set and the process is continued until all the feature vectors in the training set have been classified. The following error rates are computed from all of these classifications:

$$\begin{aligned}
false\,positive\,rate &= \frac{\#\,of\ negative\ examples\ classified\ as\ positive}{\#\,of\ negative\ examples}\\
false\,negative\,rate &= \frac{\#\,of\ positive\ examples\ classified\ as\ negative}{\#\,of\ positive\ examples}\\
true\,negative\,rate &= \frac{\#\,of\ negative\ examples\ classified\ as\ negative}{\#\,of\ negative\ examples}\\
true\,positive\,rate &= \frac{\#\,of\ positive\ examples\ classified\ as\ positive}{\#\,of\ positive\ examples}\\
overall\ error\ rate &= \frac{\#\,of\ examples\ classified\ correctly}{total\ \#\,of\ examples}
\end{aligned}$$

These error rates are used in various ways to analyze the data, such as when generating ROC curves.

8.1.2 Receiver Operating Characteristic (ROC) Curve

ROC curves were first used in World War Two for the analysis of radar signals in order to determine detection cut-off points for different operators of the radar equipment [62; 58]. When adjusting a radar, if the sensitivity is increased, then there will be more blips on the radar (i.e. false alarms) that are not necessarily interesting but the radar will be more likely to detect interesting targets. On the other hand, if the sensitivity is decreased, then there will be less distracting blips on the radar but interesting targets on the radar will also be less likely to show up. A radar's ability to distinguish between targets of interest and unwanted noise is defined as its signal-to-noise ratio [1]. Examples of noise might include birds, ground, buildings, or by radar countermeasures such as chaff (e.g. small pieces of metal deployed to distract radar-guided missiles) [1].

In terms of classifiers, the ROC curve is a plot of the sensitivity vs. (1 - specificity) for a binary-classifier as its decision threshold is varied [20]. The sensitivity can be represented by the true positive rate while (1-specificity) can be represented by the false positive rate [20]. The Equal Error Rate (EER) is calculated to represent a balance between sensitivity and specificity. In other words, it is the point at which both false accept and false reject error rates are equal [75]. Although only classifiers with K=1, and thus a single possible threshold, were used in this thesis, ROC curves are used to further analyze the training data and to compare the different scales of training data. ROC curves are also used to determine how adjustable-threshold classifiers would compare to each other in response to different training data scales.

8.2 Training Results

8.2.1 Initial Classifier (Single-Scale Training Data

Training was performed using the single-scale training data with a single VHPD classifier using K=1, $\mu = 220$, and $\tau = \infty$ (i.e. contours larger than 220 were classified). The VHPD classifier yielded the results shown in Table 8.1 when LOOCV was performed.

Training Results	
Overall Error Rate	5.02%
FP Rate	4.61%
FN Rate	6.08%
TN Rate	95.40%
TP Rate	93.93%
# Positive	1284
# Negative	3214

Table 8.1: Results from performing LOOCV on the single-scale training data with K=1 and an area threshold of 220 pixels.[28]

The values in Table 8.1 seem to indicate that the VHPD classifier performs very well, despite the partial occlusions and other difficult examples in the training data, having an overall error rate of 5.02%. Even though these error rates are low, it is not clear from them whether or not they yield a stable separation of human and non-human contours in the training data. In order to determine how much separation actually exists

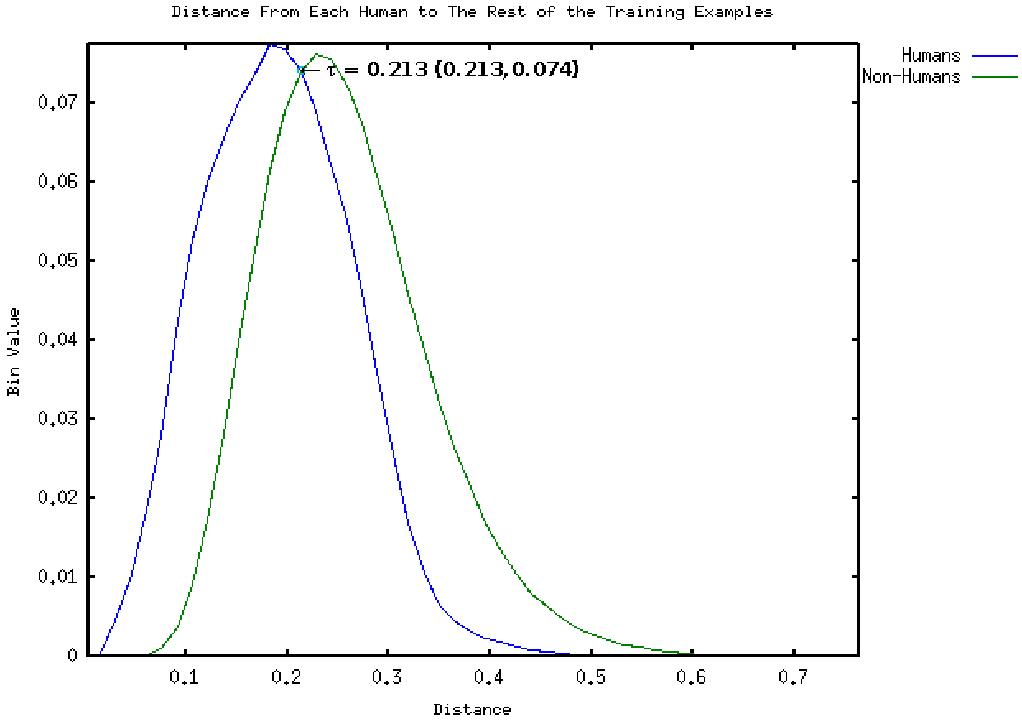

Figure 8.1: This figure shows two normalized histograms of distances from each human training example to all training examples of the VHPD classifier calculated via LOOCV. The blue histogram represents the distances from each human example to all other human examples and the green histogram represents the distances from each human example to all non-human examples. The EER is labeled as τ.

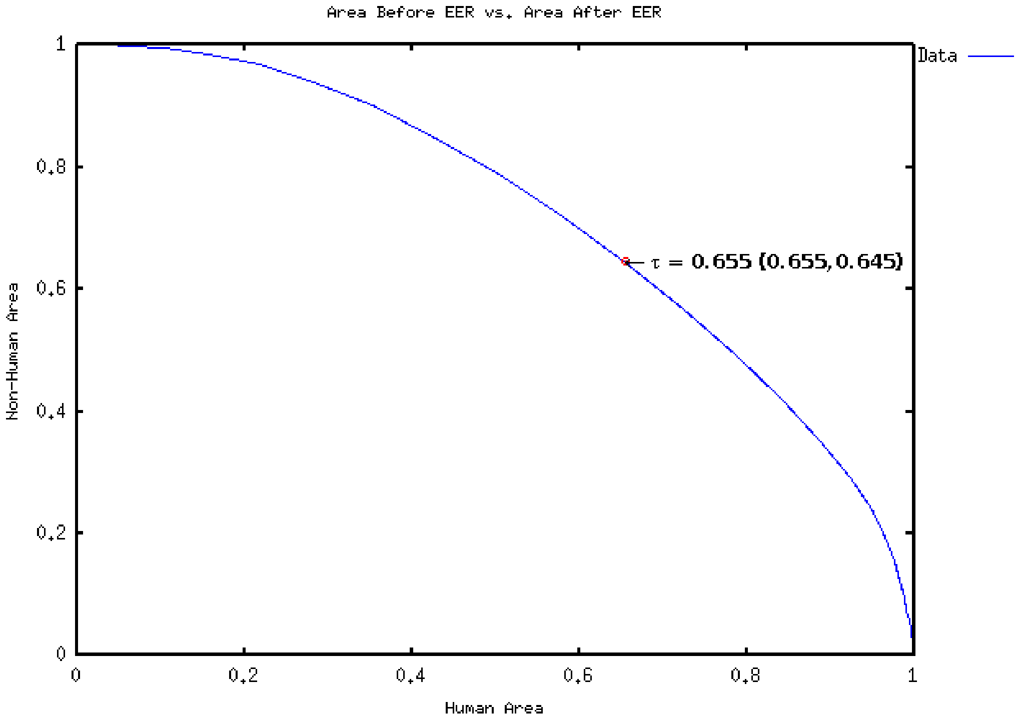

Figure 8.2: This figure shows an ROC curve of the histogram of distances from Figure 8.1 using the AUC representation. The value of τ is varied along the x-axis in Figure 8.1 and the total area under the histogram of non-human distances to the left of τ is graphed with respect to the total area under the histogram of human distances to the right of τ. The value of τ where the EER of Figure 8.1 is found is labelled.

between the positive and negative training examples, two histograms were created by calculating the distance from each positive training example to all other training examples using LOOCV, as shown in Figure 8.1. Each histogram was constructed using 50 bins with a lower bound of 0 and an upper bound of 1.0 since the distance between feature vectors is restricted to that range by the Chi-Squared metric, as described in Chapter 6. The blue histogram shows the distances from each positive training example to all other positive training examples, excluding itself (hence, LOOCV). The green histogram shows the distances from each positive training example to all negative training examples.

It is clear from these histograms that even though the error rates from Table 8.1 are low, the classifier cannot necessarily be expected to yield stable error rates. The large overlap of histograms shows that with respect to all the examples in the training set, many of the positive training examples are closer to negative training examples than they are to other positive training examples. The low error rates in Table 8.1 thus seem to imply that for each of the training examples, there is at least one training example in its class which is its closest match 95.40% of the time for negative examples and 93.93% of the time for positive examples. The results in Table 8.1, when compared to the histograms in Figure 8.1, therefore cannot be expected to hold for larger values of K as the probability that training examples outside of a given feature vector's class will match closer than training examples within the feature vector's class will likely increase as more training examples are considered.

An ROC curve was constructed from the histograms in Figure 8.1 in order to show what sort of error rates a classifier would achieve if it had the sort of overlap that Figure 8.1 has, using the Area-Under-the-Curve (AUC) representation [20]. Conceptually, the ROC curve was created by conceptually sweeping a threshold τ across the histograms in Figure 8.1 and comparing the area under the histogram of human distances to the left of τ to the area under the histogram of non-human distances to the right of τ. This is shown in Figure 8.2. The Equal Error Rate (EER), in this case, is the point at which the area under the histogram of human distances to the left of that point is equal to the area under the histogram of non-human distances to the right of that point. The EER is labeled as τ in Figures 8.1 and 8.2. If these histograms were used as a classifier, the classifier could be expected to yield error rates (if τ were set to the EER) of 34.5% for humans and 35.5% for non-humans. This is much worse than the results in Table 8.1 but it provides a much better idea of how well the polar histogram features are able to distinguish between the human and non-human contours in this training set. With this, future methods can be scrutinized more thoroughly and their classification ability hopefully more accurately predicted.

8.2.2 Multiple Classifiers (Multi-Scale Training Data)

Training was performed using the multi-scale training data with three different classifiers for the small, medium, and large image categories, respectively, using a value of K=1. The training parameters and results from performing LOOCV are shown in Table 8.2. The values of μ and τ for the lower and upper area threshold boundaries, respecitvely, were chosen by trial-and-error so that a contour around the head or face regions of the human in each positive training image would be extracted for the appropriate scales. The number of training examples corresponds to the contours with a pixel area in $(\mu, \tau]$ extracted from the training images – not to the training images by themselves.

The values in Table 8.2 indicate that several different VHPD classifiers trained with the multi-scale training set can learn that training set very well. The histograms of distances calculated via LOOCV in Figures 8.3, 8.4, and 8.5 show much better separation than the classifier trained on the single-scale training set. The AUC ROC curves corresponding to these histograms are much better as well, as shown in Figure 8.6. The fact that the AUC ROC curves in Figure 8.6 exhibit increasingly lower error rates as the scale increases implies that perhaps these classifiers, (or perhaps others in their place) when combined with the robot's motion to perform the HPD-PR method, will achieve increasingly better detection rates as the robot moves closer to a suspected human. The results of simulating the HPD-PR method with the current set of classifiers are shown in the next section.

a	S	M	L
Overall Error Rate	2.34%	2.38%	1.43%
FP Rate	2.40%	2.48%	1.60%
FN Rate	1.59%	2.38%	0%
TN Rate	97.60%	97.52%	98.40%
TP Rate	98.41%	97.62%	100%
# Positive	126	126	126
# Negative	1456	1009	1065
μ	50	147	401
τ	146	400	1500

Table 8.2: Results from performing LOOCV on the multi-scale training data for three different classifiers corresponding to the three different image scale categories of small, medium, and large with K=1

8.3 Experiment: Simulating Robot Motion Compensation With Multiple Classifiers (Multi-Scale Training Data)

8.3.1 Results

The classifier trained on the multi-scale training data, as shown in the previous section, with K=1 was executed on all nine testing series. Although theoretically the contours extracted from images taken at increasingly closer distances should achieve greater accuracy as the robot moves closer to a human, this experiment tests the classifier to see how well its performance with the training data abstracts to novel data and where its strengths and weaknesses are. The results are shown in Table 8.3. The results in Table 8.3 show the image numbers with a corresponding number 1 if at least one blob on the human in the image was correctly classified as human and a 0 otherwise. A star

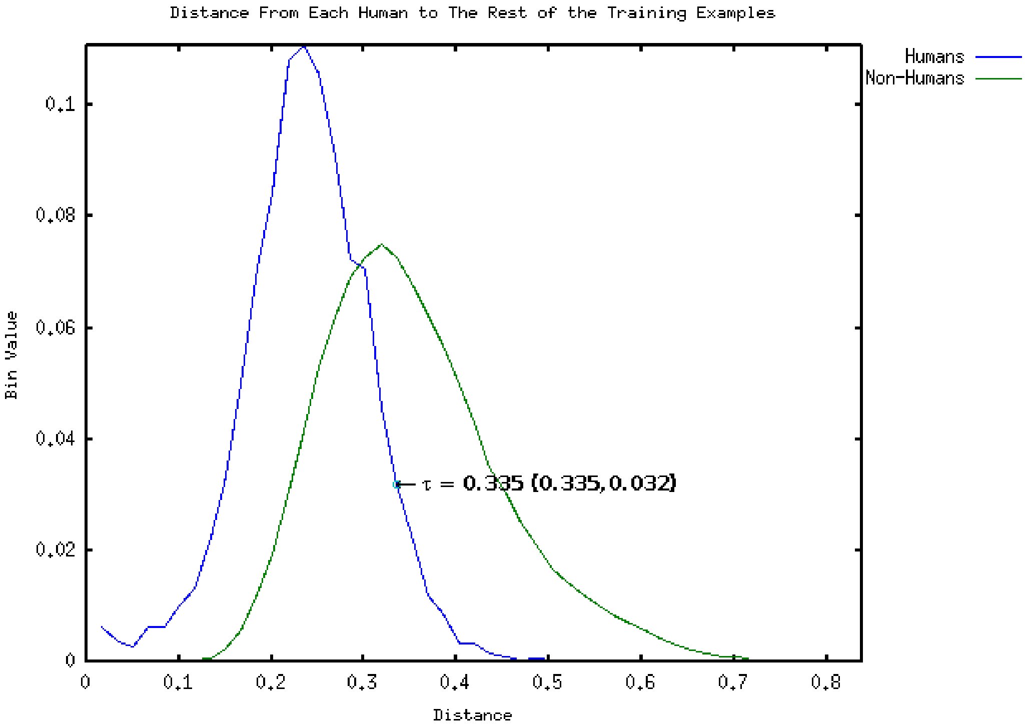

Figure 8.3: This figure shows two normalized histograms of distances from each human training example to all training examples of the VHPD classifier calculated via LOOCV using the small-scale training data. The blue histogram represents the distances from each human example to all other human examples and the green histogram represents the distances from each human example to all non-human examples. The EER is labeled as τ.

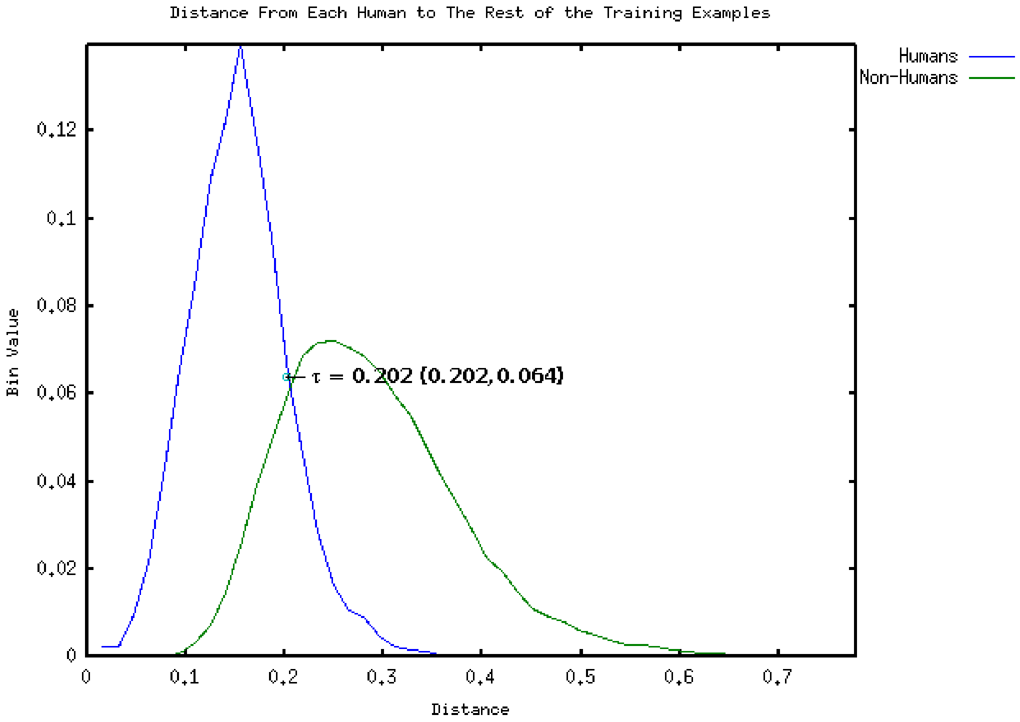

Figure 8.4: This figure shows two normalized histograms of distances from each human training example to all training examples of the VHPD classifier calculated via LOOCV using the medium-scale training data. The blue histogram represents the distances from each human example to all other human examples and the green histogram represents the distances from each human example to all non-human examples. The EER is labeled as τ.

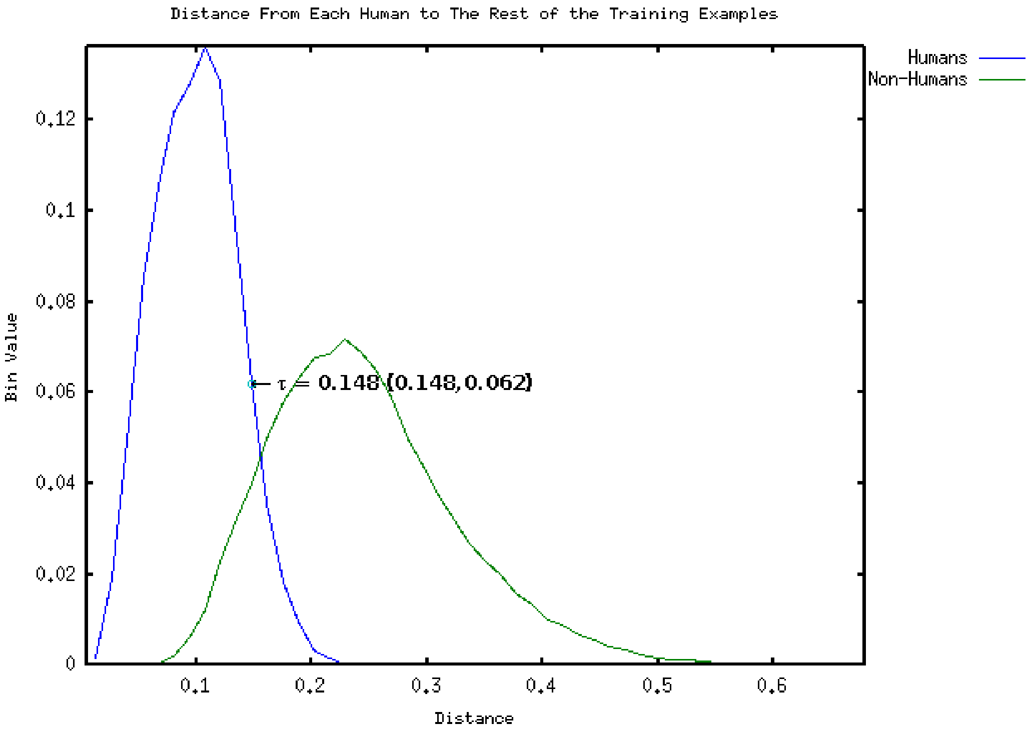

Figure 8.5: This figure shows two normalized histograms of distances from each human training example to all training examples of the VHPD classifier calculated via LOOCV using the large-scale training data. The blue histogram represents the distances from each human example to all other human examples and the green histogram represents the distances from each human example to all non-human examples. The EER is labeled as τ.

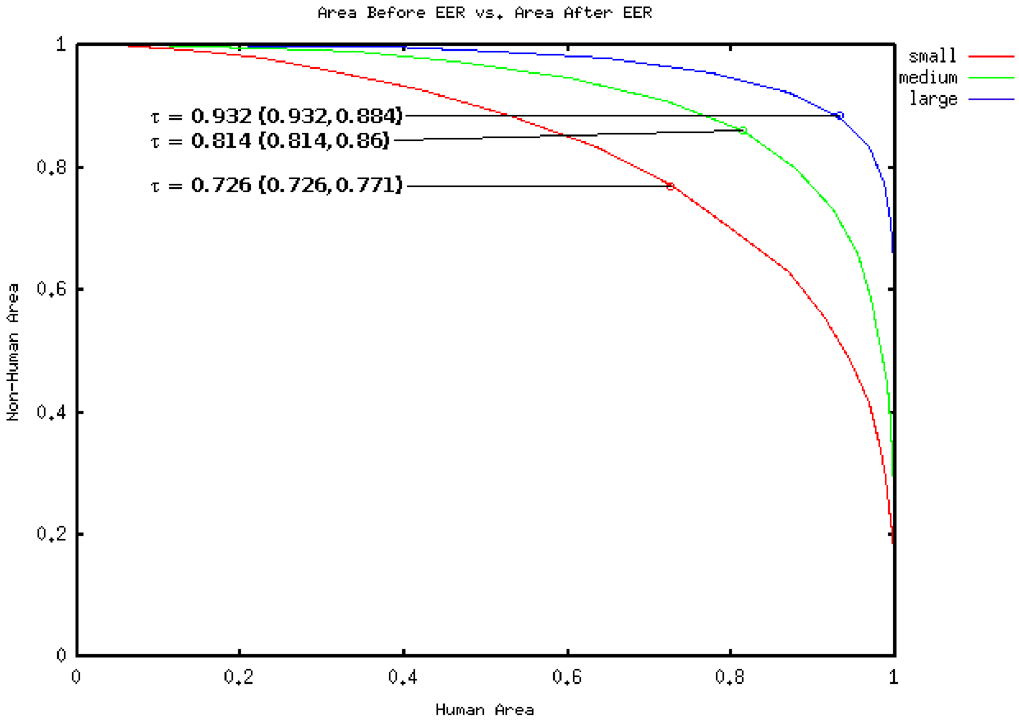

Figure 8.6: This figure shows three different AUC representations of ROC curves for the histogram of distances from Figures 8.3, 8.4, and 8.5 corresponding to the small, medium, and large image scales, respectively. The EER is labelled and was computed in the same manner as Figure 8.1. Notice how the ROC curves are progressively better as the scale of the data increases (i.e. the human is closer to the camera).

is shown by the 1 or 0 if that image included a false-positive classification.

8.3.2 Analysis

Out of all of the testing images, only two false-positive contours emerged, which are shown in Figures 8.7 and 8.8. The true-positive rate, however, was also very low. There are many possible reasons for this, some of which will be examined. One possible reason is that the heat signatures for the different image testing series are not consistent, yielding contours which do not match with the positive data. For example, image 16 of series 1 exhibited a different temperature in the nose and cheek regions which caused the face of the human to be segmented into two different contours rather than just one, as shown in Figure 8.9. Even though the upper portion of the face was classified as human, this was not the case for most of the other images in the set. Since the multi-scale training images all included the segmentation of the entire face as a single contour, this might be a large contributing factor to the results.

Problems also arose when the human was not facing the camera. In series 8, for example, there was almost nothing in the image that was hot enough to yield any contours, as shown in Figure 8.10.

Another problem that arose was the presence of objects in the background blending in with the human in the foreground. This problem manifested itself most obviously in series 7 where the hot driveway blended in with the hot human, as shown in Figure 8.11. This could possibly be mitigated if the depth of the human could be discerned from the driveway through the use of a stereo camera or a LIDAR.

Although the training set includes many negative training images with background clutter, there is comparatively low variation among the positive examples. There are also far more negative images than there are positive images. The large variation among the negative training images compared to the positive training images means that when a novel contour is compared, there is a much greater chance that a negative example exists which is a close match to the novel contour. If there is not a contour in the positive set that is a closer match, due to the low amount of variation, then the contour will be classified as negative. Further experiments should involve a larger variety of positive training examples and different mechanisms, such as a wider variety of feature types or pre-processing steps, so that the training data remains very separable.

In terms of determining if moving the robot closer to the human would result in better detections, all of the series in Table 8.3 detected the human at least twice with the exception of series 3,8, and 9 which had no detections. It is not clear whether or not this would be beneficial, however, since most of the time the humans were not detected. The fact that detections were generally sporadic indicates that if not all the frames were processed, (e.g. every other frame were skipped for efficiency reasons), then it is not clear whether or not the humans would be detected at all. The previous chapter predicts that theoretically moving closer to a potential human would result in better detection results but this has not been shown in practice with this particular classifier. If new classifiers can be discovered which yield lower error rates in practice, then the idea of repositioning the robot for better detection may prove to be a valuable behavior.

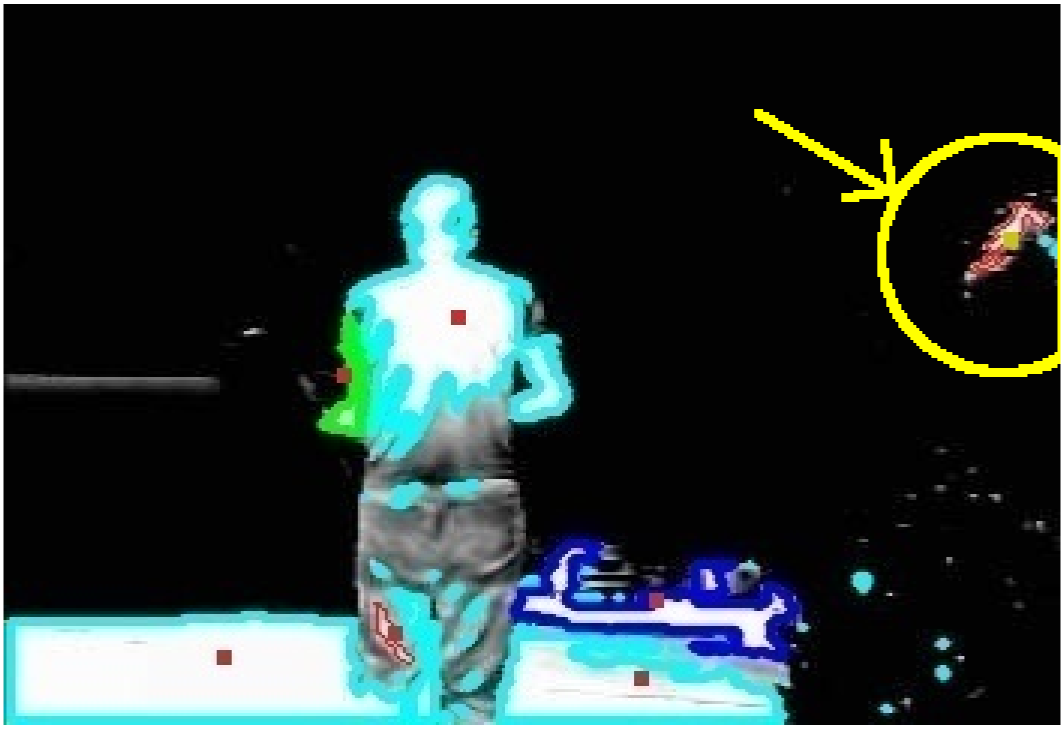

Figure 8.7: This figure shows the false-positive classification from the fifth testing series (circled in yellow with an arrow pointing to it). The red squares represent the centroids of non-human classifications and the yellow square represents the centroid of a human classification. Dark blue outlines represent contours classified by the large-scale training set, green outlines by the medium-scale training set, and red outlines by the small-scale training set. Light blue outlines represent contours whose pixel areas were either too large or too small to fall into the small,medium, or large categories so they are classified as non-human by default.

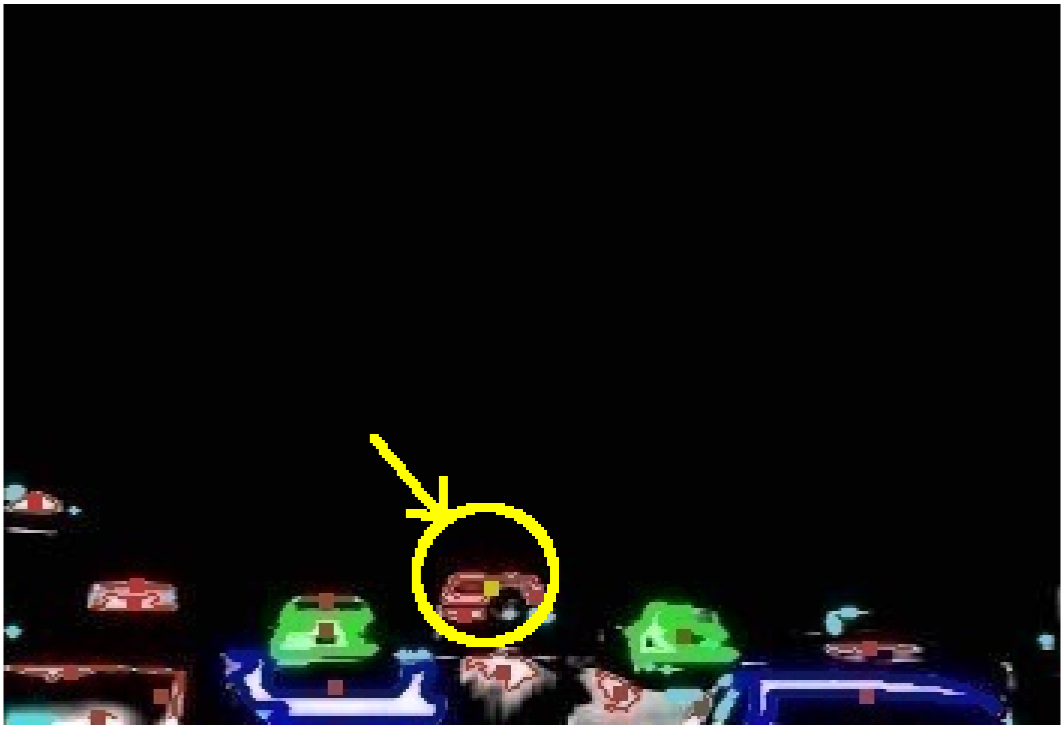

Figure 8.8: This figure shows the false-positive classification from the ninth testing series (circled in yellow with an arrow pointing to it). The color of the contours have the same representation as they do in Figure 8.7.

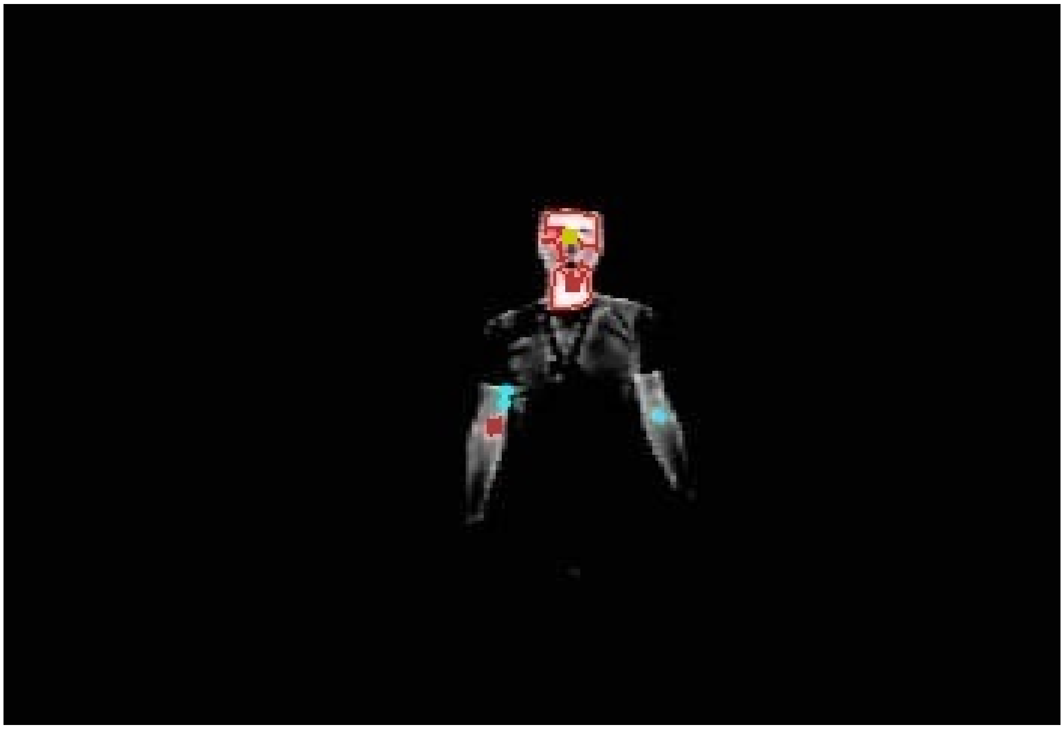

Figure 8.9: This figure shows image 16 from series 1 of the testing dataset. The face was segmented into two separate contours with only the upper contour classified as human.

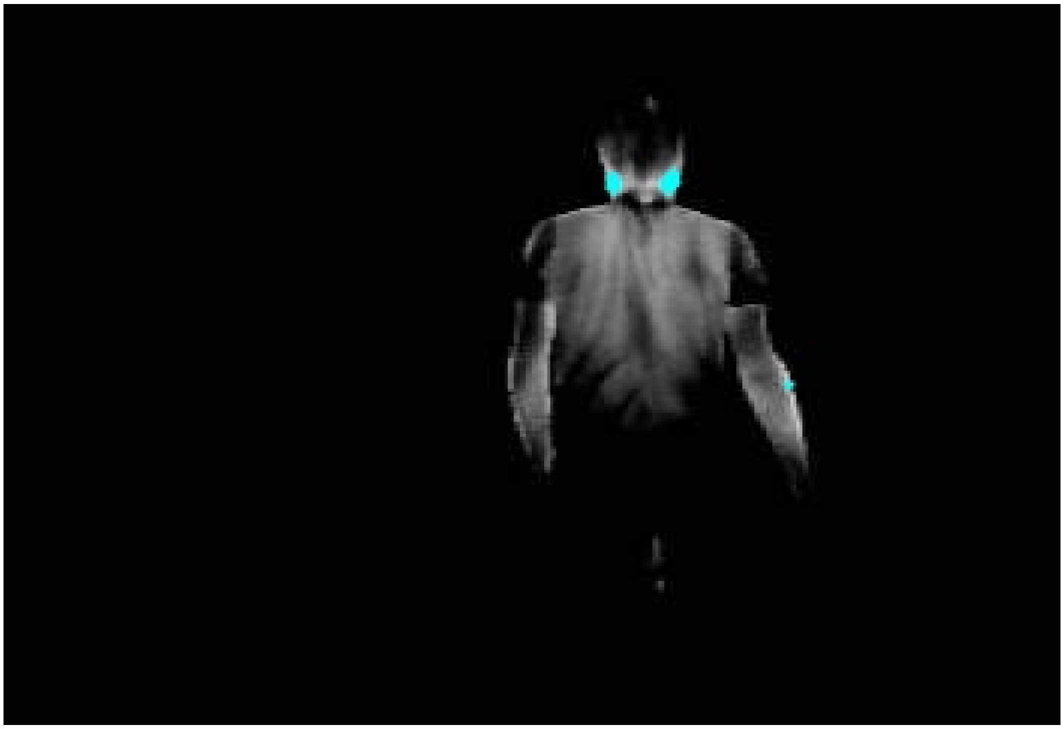

Figure 8.10: This figure shows an example of the only type of contour extracted from series 8 (using image 19) of the multi-scale positive training set. The contours are shown in light blue and are too small to be detected by the minimum pixel area threshold for the small-scale data.

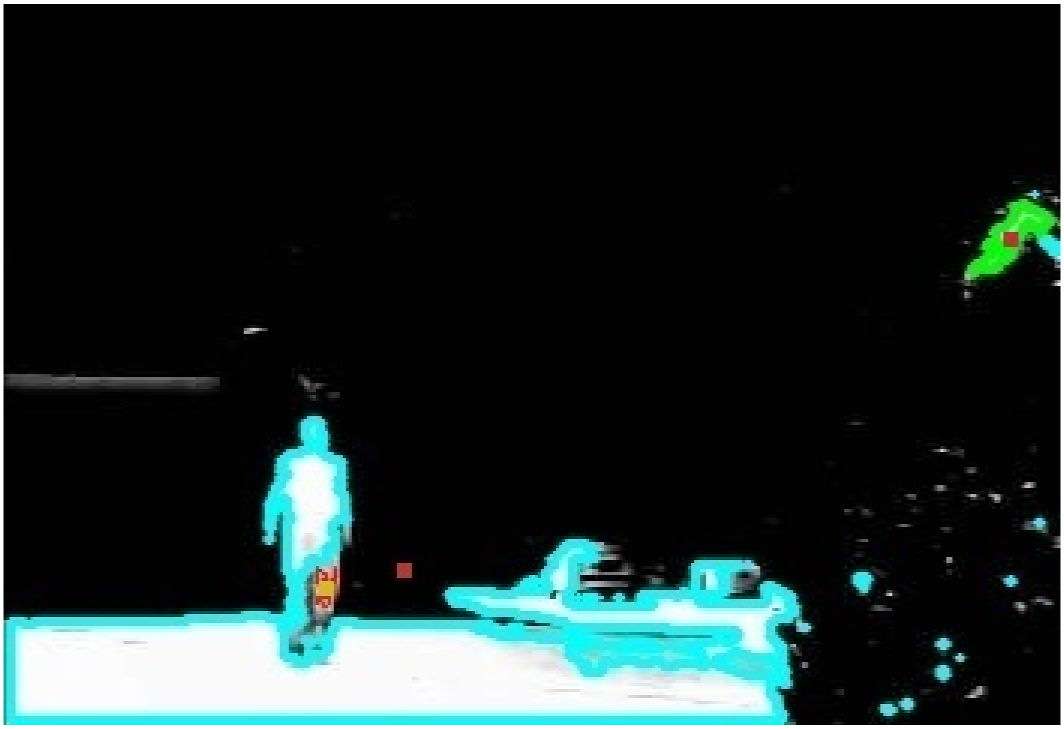

Figure 8.11: This figure shows image 9 from series 7 of the testing dataset. The human and the driveway were both hot and the contour-finding algorithm caused them to merge into a single contour , which is shown in light-blue. The back of the person's leg was classified as human though it doesn't appear to be very reliable.

image /series	1	2	3	4	5	6	7	8	9
0	1	0	0	0	0	0	1	0	0
1	0	0	0	1	0	0	1	0	0
2	0	0	0	0	1	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0
5	0	1		0	0	0	0	0	0
6	1	0		0	0	0*	0	0	0*
7	0	0		0	0	0	0	0	0
8	0	0		0	0	0	0	0	0
9	0	1		0	0	0	0	0	0
10	0	0		1	1	0	0	0	0
11	0	0		0	1	1		0	0
12	0	1		1	0	1		0	0
13	0	0			0	0		0	0
14	0	0			0	1		0	0
15	0	0			0	1		0	0
16	1				1	0		0	0
17	0				0			0	0
18	0				0			0	0
19	1				0			0	0
20					0				0
21					0				0
22					0				0
23					0				0
24					0				0
25					0				0
26									0
27									0

Table 8.3: Results from running the multi-scale classifier on the nine test series. The columns are the series numbers and the rows are the image numbers. As the image numbers increase, the human gets closer to the camera (going down the rows). The two false-positive classifications are on row 6 and are marked with a star.

Chapter 9

Conclusions and Recommendations

9.1 Summary

The history of man-portable military ground robots was examined and philosophical solutions were proposed as to how current autonomous behaviors for robots might be advanced to a deployable status. The Small Robot Technology Transfer Program at SSC Pacific was cited as an example of a currently used mechanism to hopefully expedite the development and deployment of autonomous behaviors [46]. A visual and a physical HPD method was developed starting from the available sensors and behaviors that a near-deployable version of a robot (the *PackBot*) already has. A classifier was trained on some training data containing single-scale positive training examples where the negative training examples were gathered from a real military training site and its performance was examined. Based on situations that the robot would be likely to encounter, the training data was modified so that it contained three different scales of positive training images. An additional three-stage multi-scale classifier was trained using three different image scales corresponding to rescaled versions of the initial training data where extracted contours were passed to three different sub-classifiers based on their pixel areas. The multi-scale classifier was shown to produce low error rates with the training data and increasingly better ROC curves as the image scale increased, implying that moving closer to a potential human might improve the classification rate.

The integration of the robot's motion with this multi-scale classifier was simulated using novel training data designed to push the method to its limits. These experiments showed that although the classifiers trained in this thesis are not yet accurate enough to facilitate practical detection via the use of a robot's motion, the experiments revealed various strengths and weaknesses of the camera, segmentation, and detection stages that will hopefully lead to improved classification methods.

9.2 Future Work

9.2.1 Classifiers with Adjustable Thresholds

In the same manner in which the ROC curve allows the radar operator to adapt to various situations involving cluttered environments, a classifier with an adjustable threshold can provide the same sort of flexibility to a robot. As explained in Chapter 4, a robot may encounter situations where a low false-positive rate is less important than a high true-positive rate. Using a classifier with an adjustable threshold could allow operators working with the robot to adjust its classification sensitivity to meet the requirements of the mission at hand. Some of these types of classifiers involve boosting, Support Vector Machines, or different variations on K-NN [24; 69; 6].

K-Nearest-Neighbors with Probability Estimates

For the case where $K > 1$, the posterior probability of each classified feature vector belonging to the chosen class could be estimated in addition to determining the class by a majority vote, as was mentioned by Fukunaga and Hostetler and mentioned by Atiya [25; 6]. The posterior probability is estimated by taking the number of neighbors out of the K closest and dividing that number by K. Let K_m be the number of feature vectors among the K nearest neighbors to point x that belong to class C_m. Then the posterior probability estimate is given by Equation 9.1 [6].

$$\hat{P}(C_m|x) = \frac{K_m}{K} \tag{9.1}$$

This was attempted on the training data in this thesis but was found to yield higher error rates than using $K = 1$. Other feature types and training data combinations might yield better error rates and enable a K-NN algorithm with $K > 1$ to become pragmatic.

By using K-NN for the classifier, each training example used can actually be thought of as having a weight where each weight is equal. Some training examples, however, may represent more common occurrences of a positive or negative case while other training examples may represent corner cases. By applying weights based on the training error of each match to a neighbor, the data can be better separated. This, however, would still require features capable of distinguishing between human and non-human contours very well. Atiya introduced a similar method in his paper where the weights of the K nearest neighbors were computed from data using a maximum likelihood approach [6].

9.2.2 Thresholding Methods

Other thresholding methods could be explored rather than the hard thresholding employed in this thesis , such as the histogram thresholding method introduced by Otsu or the method by Wu, et al which fuses color, infrared, and edge information to segment the foreground from the background in real-time [45; 72]. Perhaps this would allow more flexibility as to how much of the foreground and background would show up and thus hopefully make the HPD method more robust to temperature differences.

9.2.3 Segmentation Methods

When tests were run on the novel image data sets, often what should have been a single contour was instead extracted as multiple contours and what should have been separate contours were extracted as a single contour. Contours could be extracted as hierarchical edges as was done by Mhlisch, et al who compared edge images against a database of silhouettes as part of their real-time HPD method for pedestrians [43]. This could be combined with other sensory data such as depth values from a stereo camera or distances from a LIDAR [45; 55; 43; 38]. While the robot is generating maps using SLAM, perhaps humans could be detected by segmenting them from the SLAM map.

In order to extract additional information, the color camera on the *PackBot* could be combined with data from the infrared camera, as Kogut, et al did in their paper [38]. Even though it is difficult to precisely align two separate cameras, the color camera on the *PackBot* is fixed in the forward-looking direction so it doesn't need to be panned and tilted to line up with the infrared camera. If the infrared camera were deployed on a *PackBot*, it would probably be mounted in a fixed position so that it wouldn't be as susceptible to damage – making it possibly easier to align its images with the color camera.

9.2.4 Feature Types

Feature types that have been used to detect humans in color image data, such as those tested by Dollár ,et al, could be applied to infrared data, as was done by Zhang, et al with *Edgelets* and HOG features [17; 74; 71]. These features include *Shapelets* and HOG features ,which operate on gradients, or generalized Haar Wavelets [49; 12; 16]. These could also be applied to color image data and combined with the results of applying them to the infrared data, as Krotosky and Trivedi did in their paper [39].

9.2.5 Training Data

Some sort of dimensionality reduction method, such as Principal Component Analysis (PCA), could be applied to the feature vectors in order to enable more complex and yet more descriptive features to be used without making the runtime of the HPD algorithm prohibitive, as was done by Deegalla and Boström who compared several different dimensionality reduction methods on microarray data [35; 15]. The training data could also be pruned so that less relevant examples could be removed automatically, as was done by Angelova, et al [5]. Other methods could be applied to the training data such as bootstrapping or artificially generating additional training examples.

9.2.6 Context

The context of the robot could also be explored so that it ”knows” when it is outside or inside and how hot the temperature is. Perhaps the robot could measure the ambient temperature and use that to determine how much it should use the infrared camera and how much it should use another sensor input such as LIDAR. Visual cameras already adjust to the brightness of a room so perhaps this could be extended as well to minimize the importance of a particular sensor's data. Various context-specific algorithms could be integrated into a higher-level decision framework as was done by Lombardi and Zavidovique for pedestrian detection [40].

9.2.7 Realistic Simulations

Development time is a major factor when it comes to implementing and testing various algorithms. To help reduce development time, it is often very important to develop realistic simulations that can be run without the aid of a robot. An example of simulation frameworks for robots are Stage and Gazebo which interface with Player to simulate 2-D indoor environments and 3-D outdoor environments, respectively [26]. These work well for simulating robot behaviors that involve simulated objects, but there is not much room for computer vision test work since real images have a large amount of variation.

Access to robots in order to run experiments may also be limited. Deployed soldiers, for example, don't each have their own personal robot – each team that uses robots might have one robot to be shared within a small group. Similarly in research, not every scientist has his or her own robot to experiment with – they must be shared. In order to determine as early as possible whether or not a behavior will work when it is introduced to a robot, it should be simulated in a way where this can be determined with confidence.

9.3 Conclusion

The main contribution of this thesis is in outlining a philosophical roadmap to deploying autonomous behaviors on small mobile man-portable military ground robots and by exemplifying a piece of this roadmap through the development of a HPD method that utilizes the robot in a ”holistic” approach. As technology changes, particularly hardware technology, the options to explore when it comes to putting autonomous behaviors, such as HPD, on a robot will undoubtedly increase. By considering behaviors using the entire robot, hopefully the utilization of such technology will lead to more and more deployable behaviors.

References

[1] Ieee standard radar definitions. *IEEE Std 686-1997*, pages –, Mar 1998.

[2] W. Abd-Almageed and L. Davis. Human detection using iterative feature selection and logistic principal component analysis. pages 1691–1697, May 2008.

[3] F. C. Adams and G. Laughlin. A dying universe: the long-term fate and evolution of astrophysical objects. *Rev. Mod. Phys.*, 69(2):337–372, Apr 1997.

[4] G. Ahuja, D. Fellars, G. Kogut, E. R. Pacis, B. Sights, and H. Everett. Test results of autonomous behaviors for urban environment exploration. SPIE Proc. 7332: Unmanned Systems Technology XI, Orlando, FL, 13-17 April, 2009.

[5] A. Angelova, Y. Abu-Mostafa, and P. Perona. Pruning training sets for learning of object categories. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 494–501, Washington, DC, USA, 2005. IEEE Computer Society.

[6] A. F. Atiya. Estimating the posterior probabilities using the k-nearest neighbor rule, 2004.

[7] Baku13. File:sdkfz302elektr.jpg [online]. 07 August 2005. Available from: http://en.wikipedia.org/wiki/File:Sdkfz302elektr.jpg [cited 14 May 2009].

[8] M. Bertozzi, A. Broggi, M. Felisa, S. Ghidoni, P. Grisleri, G. Vezzoni, C. H. Gómez, and M. D. Rose. *Multi Stereo-Based Pedestrian Detection by Daylight and Far-Infrared Cameras*, pages 371–+. 2009.

[9] Boston Dynamics. Bigdog - the most advanced rough-terrain robot on earth [online]. 2009. Available from: http://www.bostondynamics.com/robot_bigdog.html [cited 14 May 2009].

[10] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, Jan 1967.

[11] H. Dahlkamp. Selfsupervised monocular road detection in desert terrain. In *In Proc. of Robotics: Science and Systems (RSS*, 2006.

[12] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. volume 1, pages 886–893 vol. 1, June 2005.

[13] T. Damarla. Performance of sensors mounted on a robotic platform for personnel detection. volume 6943, page 694309. SPIE, 2008.

[14] David Crane. Robo-soldier ready for combat deployment to iraq for urban warfare/ci ops [online]. 15 March 2005. Available from: http://www.defensereview.com/robo-soldier-ready-for-combat-deployment-to-iraq-for-urban-warfareci-ops/ [cited 14 May 2009].

[15] S. Deegalla and H. Boström. Classification of microarrays with knn: Comparison of dimensionality reduction methods. In *IDEAL*, pages 800–809, 2007.

[16] P. Dollár, Z. Tu, H. Tao, and S. Belongie. Feature mining for image classification. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[17] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009.

[18] H. Everett. *Unmanned Systems of WWII: Air, Land, and Sea*. draft copy, expected publication Fall 2009.

[19] H. Everett, E. Pacis, G. Kogut, N. Farrington, and S. Khurana. Towards a warfighters associate: Eliminating the operator control unit. SPIE Proc. 5609: Mobile Robots XVII, Philadelphia, PA, October 26-28, 2004.

[20] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861 – 874, 2006. ROC Analysis in Pattern Recognition. Available from: http://www.sciencedirect.com/science/article/B6V15-4HV747X-1/2/c1653cca4db4e94215437a482fcbecbb.

[21] D. Fellars, G. Kogut, G. Ahuja, B. Sights, E. Pacis, and H. Everett. Enhancing man-portable robot functionality through integration of new sensor payloads. AUVSI Unmanned Systems North America, San Diego, CA, June 10-12, 2008.

[22] FLIR Systems. Uncooled camera photon 320 [online]. 2009. Available from: http://www.corebyindigo.com/photon/photon320.cfm [cited 14 May 2009].

[23] Foster-Miller, Inc. Talon family of military, tactical, eod, maars, cbrne, hazmat, swat and dragon runner robots [online]. Available from: http://www.foster-miller.com/literature/documents/TALON-Brochure.pdf [cited 14 May 2009].

[24] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *European Conference on Computational Learning Theory*, pages 23–37, 1995.

[25] K. Fukunaga and L. Hostetler. k-nearest-neighbor bayes-risk estimation. *Information Theory, IEEE Transactions on*, 21(3):285–293, May 1975.

[26] Geoff Biggs and Toby Collett and Brian Gerkey and Andrew Howard and Nate Koenig and Jordi Polo and Radu Rusu and Richard Vaughan. The player project [online]. 30 July 2008. Available from: http://playerstage.sourceforge.net/ [cited 14 May 2009].

[27] G.Kogut, F.Birchmore, E. Pacis, and H. Everett. Using advanced computer vision algorithms on small mobile robots, 2006. SPIE Proc. 6230: Unmanned Systems Technology VIII, Defense Security Symposium, Orlando, FL, April 17-20.

[28] Glenn Mazur. Theory of inventive problem solving (triz) [online]. 26 February 1996. Available from: http://www.mazur.net/triz [cited 14 May 2009].

[29] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien. Animating human athletics. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 71–78, New York, NY, USA, 1995. ACM.

[30] iRobot Corporation. irobot packbot [online]. 2009. Available from: http://www.irobot.com/sp.cfm?pageid=171 [cited 14 May 2009].

[31] iRobot Corporation. irobot packbot 500 mapping kit [online]. 2009. Available from: http://www.irobot.com/sp.cfm?pageid=137 [cited 14 May 2009].

[32] iRobot Corporation. irobot packbot with mapping kit [online]. 2009. Available from: http://www.irobot.com/filelibrary/pdfs/gi/robots/iRobot_PackBot_Mapping.pdf [cited 14 May 2009].

[33] Jeff Schneider. Cross validation [online]. 07 February 1997. Available from: http://www.cs.cmu.edu/~schneide/tut5/node42.html [cited 14 May 2009].

[34] John Pike. Military operations on urban terrain [mout] [online]. 27 April 2005. Available from: http://www.globalsecurity.org/military/ops/mout.htm [cited 14 May 2009].

[35] I. T. Jolliffe. Principal component analysis. Springer-Verlag, 1986.

[36] Justin Pope. Pvt. robot reporting, sir! [online]. 3 February 2003. Available from: http://the.honoluluadvertiser.com/article/2003/Feb/03/mn/mn01a.html [cited 14 May 2009].

[37] Kerolyn1313. File:cross-sectional microbolomter.jpg [online]. 3 December 2007. Available from: http://upload.wikimedia.org/wikipedia/en/b/bb/Cross-sectional_microbolomter.jpg [cited 14 May 2009].

[38] G. Kogut, G. Ahuja, B. Sights, E. Pacis, and H. Everett. Sensor fusion for intelligent behavior on small unmanned ground vehicles, 2007. SPIE Proc. 6561: Unmanned Systems Technology IX, Defense & Security Symposium, Orlando, FL, April 9-13.

[39] S. Krotosky and M. Trivedi. Person surveillance using visual and infrared imagery. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(8):1096–1105, Aug. 2008.

[40] P. Lombardi and B. Zavidovique. A context-dependent vision system for pedestrian detection. pages 578–583, June 2004.

[41] Los Alamos National Security, LLC. Roadrunner, world's first petaflop/s system, remains #1 on the top 500 and #7 on the green 500 [online]. 2009. Available from: http://www.lanl.gov/orgs/hpc/roadrunner/ [cited 14 May 2009].

[42] S. Moon. Novel infrared absorbing material coupled uncooled microbolometer. pages 658–660 vol.2, Oct. 2004.

[43] M. Mhlisch, M. Oberlnder, O. Lhlein, D. Gavrila, and W. Ritter. A multiple detector approach to low-resolution fir pedestrian recognition. In *In Procs. IEEE Intelligent Vehicles Symposium 2005*, pages 23–28, 2005.

[44] Online Data Services Ltd. Thinking methods [online]. 2009. Available from: http://www.ideaconnection.com/thinking-methods/triz-00015.html [cited 14 May 2009].

[45] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, January 1979.

[46] E. Pacis, H. Everett, N. Farrington, and D. Bruemmer. Enhancing functionality and autonomy in man-portable robots. SPIE Proc. 5422: Unmanned Ground Vehicle Technology VI, Orlando, FL, April 13-15, 2004.

[47] D. Powell, G. Gilbreath, and M. Bruch. Multi-robot operator control unit for unmanned systems. Defense Tech Briefs, August 1, 2007.

[48] P. Rudol and P. Doherty. Human body detection and geolocalization for uav search and rescue missions using color and thermal imagery. pages 1–8, March 2008.

[49] P. Sabzmeydani and G. Mori. Detecting pedestrians by learning shapelet features. pages 1–8, June 2007.

[50] Serge Belongie. Handy equation reference for cse 166 (image processing) [online]. 25 September 2002. Available from: http://www-cse.ucsd.edu/classes/fa02/cse166/eqnsheet.pdf [cited 5 May 2009].

[51] Sgt. Lorie Jewell / Army News Service. Armed robots to march into battle [online]. 6 December 2004. Available from: http://www.defenselink.mil/transformation/articles/2004-12/ta120604c.html [cited 14 May 2009].

[52] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainly. *Int. J. Rob. Res.*, 5(4):56–68, 1987.

[53] Sourceforge, Inc. Open computer vision library [online]. 2009. Available from: http://sourceforge.net/projects/opencvlibrary/ [cited 14 May 2009].

[54] Spc. Jonathan Montgomery. Eod robots [online]. 16 January 2005. Available from: http://usmilitary.about.com/od/armyweapons/a/eodrobot.htm [cited 14 May 2009].

[55] L. Spinello and R. Siegwart. Human detection using multimodal and multidimensional features. pages 3264–3269, May 2008.

[56] SSC Pacific. Unmanned systems branch 2371 [online]. 3 June 2008. Available from: http://www.spawar.navy.mil/depts/d30/d37/d371/d371_idx.html [cited 14 May 2009].

[57] Staff Sgt. Tony M. Lindback. Soldiers can take steps to avoid explosives [online]. 21 September 2007. Available from: http://hammerslogger.gulfeast.com/index.php/hammerslogger/2007/09/ [cited 14 May 2009].

[58] StatsDirect Limited. Roc curve analysis [online]. 2008. Available from: http://www.statsdirect.com/help/nonparametric_methods/roc_plot.htm [cited 14 May 2009].

[59] Syque. Triz contradiction analysis [online]. 2008. Available from: http://creatingminds.org/tools/triz/triz_contradiction_analysis.htm [cited 14 May 2009].

[60] The Altshuller Institute for TRIZ Studies. The altshuller institute for triz studies [online]. 2009. Available from: http://www.aitriz.org/ [cited 14 May 2009].

[61] The Defense Advanced Research Projects Agency. Darpa grand challenge 2005 [online]. 30 December 2007. Available from: http://www.darpa.mil/grandchallenge05/overview.html [cited 14 May 2009].

[62] Thomas G. Tape. Interpreting diagnostic tests [online]. 2001. Available from: http://gim.unmc.edu/dxtests/Default.htm [cited 14 May 2009].

[63] United States Air Force. Global hawk [online]. October 2008. Available from: http://www.af.mil/factsheets/factsheet.asp?id=13225 [cited 14 May 2009].

[64] United States Air Force. Mq-1 predator unmanned aircraft system [online]. September 2008. Available from: http://www.af.mil/factsheets/factsheet.asp?id=122 [cited 14 May 2009].

[65] United States Department of Energy Office of History and Heritage Resources. The manhattan project: An interactive history [online]. Available from: http://www.cfo.doe.gov/me70/manhattan/events.htm [cited 14 May 2009].

[66] unknown. Goliath photos [online]. Available from: http://www.worldwar2aces.com/panzer-tank/goliath/ [cited 14 May 2009].

[67] unknown. Goliath remote controlled anti-tank vehicle [online]. Available from: http://anonymous-generaltopics.blogspot.com/2008/04/goliath-remote-controlled-anti-tank.html [cited 14 May 2009].

[68] unknown. Light demolition carrier ”goliath” (sd.kfz.302/303a/303b) [online]. Available from: http://www.geocities.com/CapeCanaveral/Lab/1167/egoliath.html [cited 14 May 2009].

[69] V. Vapnik. Estimation of dependences based on empirical data [in russian]. Nauka, Moscow, 1979. (English translation: Springer Verlag, New York, 1982).

[70] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features, 2001. Available from: citeseer.ist.psu.edu/viola01rapid.html.

[71] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. volume 1, pages 90–97 Vol. 1, Oct. 2005.

[72] Q. Wu, P. Boulanger, and W. F. Bischof. Robust real-time bi-layer video segmentation using infrared video. *Computer and Robot Vision, Canadian Conference*, 0:87–94, 2008.

[73] R. Zemeckis. Back to the future. 1985. Universal Pictures.

[74] L. Zhang, B. Wu, and R. Nevatia. Pedestrian detection in infrared images based on local shape features. pages 1–8, June 2007.

[75] Zvetco Biometrics LLC. Biometric knowledge center: Definitions [online]. 2008. Available from: http://www.zvetcobiometrics.com/Support/definitions.jsp [cited 14 May 2009].