4-16-2015 12:00 AM

Geological Object Recognition in Extraterrestrial Environments

Gregory M. Elfers
The University of Western Ontario

Supervisor
Dr. Olga Veksler
The University of Western Ontario

Graduate Program in Computer Science
A thesis submitted in partial fulfillment of the requirements for the degree in Master of Science

GEOLOGICAL OBJECT RECOGNITION IN EXTRATERRESTRIAL ENVIRONMENTS

(Thesis format: Monograph)

by

Greg Elfers

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Masters of Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Acknowlegements

I would like to acknowledge Melissa Elfers, my wife, who graciously shares me with my computer, but still loves me more than my computer ever has. Dr. Olga Veksler who probably didn't realize what she was getting into when she agreed to supervise me. My sister Cynthia, whose patience with me in this life has gone far beyond what siblings should have to endure and Stacy "Spacey" Orsat, who helped me start down the road to university. Owen McCarthy, who spent many hours beside me in undergrad, listening to me swear at code, he protected my sanity and challenged me academically even if he was never able to get that last mark. Tom Cunningham whose example showed me that any problem could be solved with patience and determination. Dr. Steven Beauchemin who gave me an opportunity to be excited about Mathematics. Dr. Taha Kawasari who was always ready to answer a question about that math, despite being eyeball deep in his own thesis. And a special thanks to Bernard Sandler who pointed me in the direction of the end even if I'm not that good with following directions.

This thesis is also dedicated to the fond memory of:

Barbara and Manfred Elfers, The height of a building, depends on the strength of its foundation.
Jane Bashara, whose encouraging light was dimmed before its time,
Dr. Sheng Yu, who showed me the beauty of computing,
Greg Mate, whose optimism during his fight against cancer put the battles I face in perspective.

Abstract

On July 4 1997, the landing of NASA's Pathnder probe and its rover Sojourner marked the beginning of a new era in space exploration; robots with the ability to move have made up the vanguard of human extraterrestrial exploration ever since. With Sojourners landing, for the rst time, a ground traversing robot was at a distance too far from earth to make direct human control practical. This has given rise to the development of autonomous systems to improve the e?ciency of these robots,in both their ability to move,and their ability to make decisions regarding their environment. Computer Vision comprises a large part of these autonomous systems, and in the course of performing these tasks a large number of images are taken for the purpose of navigation. The limited nature of the current Deep Space Network means that a majority of these images are never seen by human eyes. This work explores the possibility of using these images to target certain features by using a combination of three AdaBoost algorithms and established image feature approaches to help prioritize interesting subjects from an ever growing data set of imaging data.

Keywords: Computer Vision, Shatter Cones, Autonomous Robotics, Machine Learning

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Since December 13th, 1972, when Apollo 17 Astronauts Eugene Cernan and Harrison Schmitt took their last steps on the lunar surface, the exploration of extraterrestrial environments has been the exclusive providence of robots. In recent years with the success of Pathnder/Sojourner in 1997, the Mars Exploration Rovers (MER) Spirit and Opportunity in 2003, and Mars Science Laboratory (MSL) Curiosity on August 6th 2012, the usefulness of robotic surface rovers in extraterrestrial research has been rmly established. With a number of rover missions planned for launches in the next 10 years by the space agencies of China, India, Russia, and the European Union, the importance of these mechanized explorers cannot be understated. Current protocols require human intervention in almost every step, but growing mission demands and improving capabilities mean that rovers are covering more distance without the assistance of humans. Pathnders rover, Sojourner, for example, traveled a total of 100 meters over the entirety of its 3 month mission while Opportunity has traveled over 42.23 kilometres in its 12 years on the Martian surface (slightly more than the 42.195 kilometre distance of an Olympic Marathon[40].) With the arrival of Curiosity, and an ever growing list of long duration orbiters and deep space probes, combined with the costly equipment needed for increases in the bandwidth available in the Deep Space Communication Array, autonomy for these rovers is becoming a more apparent requirement.

The rst steps of this autonomy began with the MER rovers GESTALT navigation system which helps the rover with the minutiae of driving. The next step was taken with the upload of CalTech/JPL's AEGIS Autonomous Science software package to Opportunity in 2009. The foundation of any autonomous system rests in the ability of that system to correctly identify and di?erentiate the objects around it. The rocks on the surface of an extraterrestrial world are a rich source of information. Their location, chemical makeup, and physical structure can tell the environmental history of the body on which they sit. This is especially true in the area around craters where the violence of an impact can reveal clues otherwise hidden deep within the bedrock of a planet.

One of the largest constraints faced in the development of algorithms for this kind of work, lies in the limited computational power available to the rover. MER Opportunity has 20MHz RAD6000 Processor with 128MB of RAM and 256MB flash memory. The AEGIS system

regularly processes 1 MB images while having access to only 4MB of RAM [27]. This means that an operation which takes mere seconds on a modern processor, will take many minutes or hours on a rover. While the MSL Curiosity rover has a processor that is an order of magnitude more powerful as the one which AEGIS runs on, it is still several orders of magnitude slower than a modern laptop. While the scope of this work will not limit itself to currently available rover processing power, it is essential that our solutions are informed by such limitations.

Despite the limitations of processing power, perhaps the most compelling argument to be made, comes from the limitations in bandwidth. At the current time any spacecraft not in earth orbit communicates with earth through NASA's deep space communication network, which consists of three sights located 120 degrees apart on the planet, with stations in Spain, California, and Australia. As the capabilities of the instrumentation on each successive generation of spacecraft grows more complex, so to does the amount of data they are able to generate. Current craft such as the Lunar Reconnaissance Orbiter are expected to generate over 70 TB of data over its serviceable lifetime, Which is an order of magnitude larger than the Mars Reconnaissance Orbiter launched just a few years earlier. In order to save bandwidth current missions send low resolution image thumbnails which allows mission planners to make decisions about science targets and navigation. If mission control requires, the rover will send full resolution copies of the data on subsequent communications.

While this system works well for current missions, it means that only a small fraction of existing imagery is used. Some data, such as navigation and hazard avoidance imagery is never seen by human eyes, and of the images returned, the resolution of thumbnail data may be insufficient to capture key details contained within the rocks. Such an approach is also very slow, with the rover moving mere feet per day, transmitting imagery back to earth and then awaiting instructions on how to proceed. Ultimately it is the slow pace of this method of exploration that presents its largest drawbacks.

In its planning documents NASA has identified sample return missions as a key goal in the intermediate future. Such a mission would demand a much quicker timeline than current operating procedures allow. To maximize the effectiveness of the science on a mission of this type, a much larger area would have to be explored in greater detail more rapidly than current methods would allow. A rover with the ability to detect objects of interest in its surrounding environs would offer a great advantage in a mission such as this.

1.2 Scope of our investigation

In our investigation, we are looking at two proof of concept implementations of a method of identifying a terrain of interest, that will demonstrate the feasibility of developing generalizable feature descriptors to allow for a general image description language. Such a language would allow for a flexible method of allowing a rover to perform an unsupervised classification of specific terrain types of interest to ground based controllers. This work will incorporate a variety of image feature types despite the current limitation is extraterrestrial processing power, with the belief that such a system may be useful if the techniques investigated allow for such a system to be incorporated into the data collection methods used in the field of Terrestrial Geology.

We have decided to limit our investigation to macroscopic features as this does not require

special equipment or close investigation by a rover. This would allow for a program to run in conjunction with existing navigational and scientific operations, to increase the potential of scientific discovery without changing primary mission perimeters. This work will focus on two features at the scale of "Field Geophotography." This is the scale of photos used by field geologists in their work and limit its scope to images capable of being photographed by a standard commercial point and shoot camera. For the purposes this work, this means image frames representing a distance of 20cm-100m Large landscapes will not be included except where they form the background of closer objects.

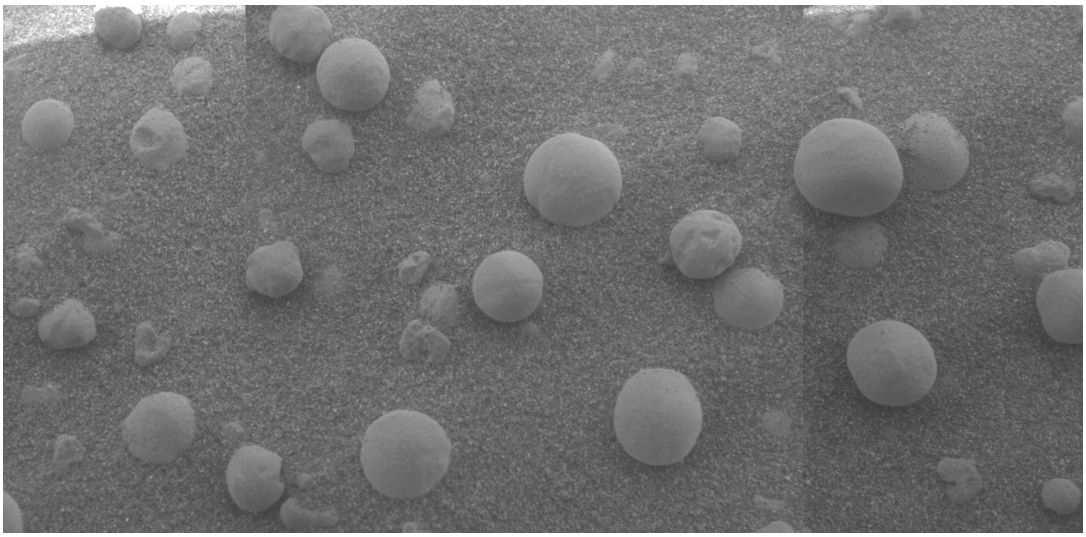
The features chosen for investigation are outcrops and shatter-cones. Outcrops were selected because of they represent the underlaying structure of a geological region, and may give clues as to the conditions that existed during their formation. Such structures may also be of interest in the identification of global phenomenon. And the presence of certain minerals, their deposition and erosion can give clues as to the historical conditions over a broad timescale. Such features exist with relative frequency and will almost certainly exist within the landing zone of a rover. Besides the scientific interest provided by the make up of individual examples, they are also useful as landmarks. A change in appearance of local outcrops can delineate different geological zones, or their orientation and arrangement within a landscape can suggest larger underlaying structures.

The second feature, shatter-cones, provide a unique opportunity of automated investigation for a number of reasons. A shatter-cone is unique to a specific range of impact events, and present at the same time, quite unique and yet almost universal features in their appearance. They are indicative of a specific, set of pressures present at the time of their creation, and have been found in a variety of rock types in a number of locations on earth. As of yet these structures have not been identified on other planets, and since most imagery is returned to earth in a low resolution format that might obscure their characteristic features, they represent an ideal candidate for automated detection from a rover. Shatter-cones also represent a tantalizing terrestrial application for locating impact craters here on earth. The Earth Impact Database maintained at the University of New Brunswick identifies only 188 confirmed impact sites, which is a fraction of the sites observed on other planets and moons.[52] The unique and macroscopic nature of this feature makes it a good candidate for identification in existing images and the mining of online photo repositories could offer clues of undiscovered impact sites.

1.3 Approach

While object detection is one of the most active fields of study in computer vision, it cannot be thought of as a routine or common place process. Very often the most successful algorithms depend on applying situational heuristic tuning or on identifying descriptors that fit well with specific features of the object being sought. An approach to face detection can be reasonably certain that the overarching design and main structural features of a human face do not vary greatly from person to person. Likewise, despite the variances in the design of cars, certain operational requirements mean that similar looking structures can be found in predictable locations in an overwhelming number of the various iterations of cars.

One of the biggest difficulties in the autonomous detection of geological features, is that



Credit: *NASA*

Figure 1.1: The existence of Hematite spherules found embedded in rock would illicit far less comment on earth than it did when Opportunity found these spherules at Meridiani Planum in 2004. Such structures are strong indicators of the presence of liquid water during their formation.

most of the parameters which might be used to describe an object can vary to an extraordinary degree. Objects with almost identical chemical makeups can vary widely in appearance. Likewise completely dissimilar objects can appear almost identical to one another on all but the microscopic level. While the morphology of a region can impose some structural similarities in appearance between different rock features, there are can be large variations due to the chemistry of the surrounding rocks. To complicate things further, the context of a feature within or with respect to its environment or even within the context of similar structures elsewhere on a planet, can change its scientific interest significantly. For example, similarities in the fossils and minerals in rocks in Africa and South America suggested that these two distant locations once touched, and an image of hematite spherules like the one in Figure 1.1 would have elicited little interest were they to have been found on Earth rather than on Mars.

In light of the complexities of this problem, and with the expectation of the availability of human expertise in the planning and execution of such missions, we have opted to develop a system in which the objects of interest are known at least to some degree,and can be pre-trained from a set of known examples. Or with the help of human expertise a training database can be constructed with a modestly sized subset of gathered examples To this end we have chosen to use the machine learning algorithm AdaBoost to train a classier from a set of feature vectors created from weak descriptors in order to create a decision making committee that is able to identify objects of interest. As we will see in our discussion in Chapter 5, a machine learning approach o?ers a number of key advantages over other methods.

While both outcrops and shatter-cone features will involve the same approach and work flow, the creation of our database will differ in the selection of source imagery used to construct the training data. In the case of outcrops we will try to use a subset of images gathered over a two week time period and apply our identification mechanisms to unique and novel images

from the same geological region but taken of different structures or from different points of view. With the shatter-cone training data we will use examples of shatter-cones from a single geographic location and attempt to apply our detection mechanisms to similar structures in widely different geological regions.

1.3.1 Program Workflow

Data Set preprocessing The process will begin by building a data set from our test images isolating just those features which are to be identified by the algorithm, a roughly equal number of negative examples from those same images will also be created. Each image in the data set will be resized to 100x100 pixels to represent different resolutions, then a histogram normalization algorithm is applied to the images and a greyscale version created. The appropriate image is then passed to each of the feature extractors.

Feature Extraction For the purpose of our investigations 9 different features were isolated from 7 different descriptor strategies we extracted from the image. The feature set consists of:

Scale Invariant Feature Transform(SIFT)[42]
Haar like features (Haar)[63]
Histogram of Oriented Gradients(HOG)[19]
Edge Orientation Gradients(EOG)[29]
Hough Transforms[23]
Intensity Histograms
Edge Density

Machine Learning Once the features are extracted a feature vector will be constructed and labeled as either an interesting example or an uninteresting example at which point a boosting algorithm[57] will be used to create a decision making committee to evaluate new images.

Object Identification To identify features within a novel object, a feature vector will be created from sub-windows of the image we are analyzing and a probability map of the image as a whole will be created by calculating the decision making committee's evaluation of each sub-window.

1.4 Outline

This thesis will be organized as follows: In Chapter 2 we will give an overview of approaches used by researchers studying similar problems. In Chapter 3 we look at the specific geology we have selected to test our method on while In Chapter 4 we will go over the image features we have chosen to use, and how we constructed them. In Chapter 5 we will give an overview of various machine learning techniques and examine the AdaBoost Algorithm in some detail. Starting with Chapter 6 we will begin a look at the specifics of our work with the building and refinement of our data sets. In Chapter 7 we will review the details of our features for the

chosen geology, and discuss our results. In Chapter 8 we will conclude with a discussion of the implications of our results.

Chapter 2

Previous Work

The use of computer vision algorithms in the field of Geology is a rapidly growing practice. Starting in the defence and space industry for use with satellite imagery, and more recently spurred on in large part by the resource extraction and processing industries who have benefited from technologies developed for the military's increasing use of unmanned aerial vehicles (UAVs) automated surveying through a variety of optical and infrared techniques forms an increasingly important part of field work. Traditionally however this work has focused on either in work on large scale aerial imaging or in the realm of the microscopic for the purpose of automated mineral identification. The reasons for this can be understood in the context of the requirements of fieldwork. Terrestrial geological exploration often happens in locations where power is limited, and access is difficult. This has meant that the logistics required to insert and power a ground based system for the purpose of autonomous geology has proved more unwieldy than useful to an individual geologist in the field. In such cases where a terrestrial location is unavailable to human exploration, the relative simplicity and reliability of teleoperation has biased remote exploration systems towards those systems. Even in Extraterrestrial environs as far away as the moon, human guided systems are much more capable than autonomous system are. It is only with the development of robotic rovers for the exploration of more distant locations such as Mars, that both the need, and operational ability have come together.

While this has been the case in the past, it is certainly not true of the future. While the rigours of space travel have demanded that computer processing power be adopted more slowly, it is now reaching a stage where reasonably complex and powerful computer systems for controlling autonomous identification systems are being launched. Likewise with the growth of low power processors, and the growing efficiency of power systems it is reasonable to assume that the trend will continue including the growth of more autonomous aspects of terrestrial field work. Much of the previous work focuses on one of three broad categories, landscape awareness for autonomous navigation, commonly referred to as "Simultaneous Localization and Map Building" (SLAM), rock finding algorithms for identifying protruding objects, and material identification algorithms for autonomous identification of rock composition.

2.1 SLAM

SLAM techniques can be quite comprehensive in scope, but can be subdivided into three general areas, position identification, obstacle avoidance, and path finding. A lot of work has been done in this area in recent years, with the most famous terrestrial examples of such vehicles being "Stanley," the winner of the "2005 DARPA Grand Challenge," and Google's "Self-driving" Car [60, 61]. Both of these vehicles make use of point clouds created by an omnidirectional Light Detection and Ranging (LIDAR) system, and require massive amounts of computational power due to the relative speed of the vehicles. While none of the current generation of rovers use LIDAR systems for navigation, the possibility is being investigated for future Rovers [2, 54]. Position identification in the context of autonomous navigation is used to locate the vehicle within its environment and to help provide more robust odometry data. Obstacle avoidance is often a realtime endeavour whose goal is to identify immediate hazards located along the direction of travel and within a relatively short range (relative to the vehicles velocity.) Pathfinding on the other hand often requires a broader, more panoramic and longer range view of the surrounding landscape and its hazards. Pathfinding can consist of both realtime and pre-plotted components choosing a prospective path between two points, and in the case of a system such as a rover, path corrections are selected as new information is provided by obstacle avoidance routines. A number of different optical techniques have been developed to this end, including land mark validation [64], Visual Odometry(VO) [44, 49], maximum likelihood matching range maps [48], and rock modelling and matching [41]. The Mars Exploration Rovers (MER) Spirit and Opportunity and now The Mars Science Laboratory (MSL) Curiosity are currently using some of these techniques to help them traverse dangerous terrain. (See section 2.4.)

2.2 Rock Detection

Rock detection is an important part of operational and scientific awareness at a given location. Depending on the required definition of what constitutes an interesting rock, or even what constitutes a rock at all for that matter, will greatly influence the choice of rock identification strategy employed. Lighting, positioning, occlusion, albedo, and contrast can all greatly influence the appearance of a scene. Some algorithms attempt to circumvent the variances by using a simplifying assumption to classify rocks. Some make assumptions about relative lightness, or limit the shape [18, 17]. While more sophisticated algorithms use a number of different image segmentation techniques including edge detection [9, 10], mean shift [16], Hue Saturation and Intensity variances [59], graphs [28], and more recently graph cuts and super pixel segmentation [34]. In cases where stereo imaging is available a number of different researchers have taken advantage of that information for segmentation [35, 58, 46].

2.3 Material Identification

To someone who can understand the signs, any given rock is much more than just a lump of coalesced minerals. Shaped by, impact volcanos, wind and water, the structures, colours,

shapes, textures, albedo, wear, patterning all give some hint to how a rock came to be where it was. Geologists have used these clues to extract the history of rocks, regions and planets from these clues. It is no surprise then that computer scientists have sought to duplicate the human Geologist's methods with computer vision. Shape recognition strategies include, convexity or outline comparison to a circle or ellipse [8, 53], the fractal dimension of a shape [36], edge distance to a centroid and Fourier analyses thereof [4, 22]. Statistical analysis of models describing forms [47] or edge curvature [25] have also been tried. Somewhat more complex than shape analysis, is texture analysis. The exact definition of what constitutes a texture is less obvious than it might seem. Some textures are structured, some are stochastic, some are periodic, and some such as text on a page, are only evident through statistical analysis. A number of different descriptors of what constitutes a texture have been tried. Chaudhuri used a measure he called the fractal dimensionality [14], another technique is to use intensity and create descriptors from a statistical analysis of a gray-level co-occurrence matrix[51, 65], Given the similarities of structures in the eye to the outcome of Gabor filters, some researchers have used, directionality histograms [37] and textons formed from responses of convolving the image with a bank of these filters [11, 13, 33, 38, 39].

2.4 OASIS, AEGIS and GESTALT

Lastly there are the three autonomous decision making systems that are currently in use on the NASA MER rovers. The first, used for assistance in navigation is known as the Grid-based Estimation of Surface Traversability Applied to Local Terrain (GESTALT) system. This system uses the stereo imaging for hazard avoidance and to improve the accuracy of the odometry data gathered from the drive system [3]. The Autonomous Exploration for Gathering Increased Science (AEGIS) system uses a rock identification program known as "Rockster" [9] to identify possibly interesting rocks or outcrops. The Rockster algorithms use edge detection to identify an object of interest and then the system positions MER's panoramic cameras to take a picture of the resulting selection. This system has been in operation on the MER rover(s) since 2009[27]. AEGIS forms one component of a larger autonomous science package known as The Onboard Autonomous Science InvestigationSystem (OASIS), in addition to the capabilities of AEGIS, OASIS segments sky from ground and looks for both geological and meteorological phenomena which might be of interest [12].

Chapter 3

Geological Features

> Rock, *n.* The solid mineral material forming much of the substance of the earth (or any similar planetary body), whether exposed on the surface or overlain by soil, sand, mud, etc.
>
> -The Oxford English Dictionary[21]

Before discussing an algorithmic approach of identifying an object that might be labeled as a "rock" on another planet, it is helpful to more clearly define the objects we propose to identify. As the definition above suggests, the word rock is a nebulous and almost meaningless word in any real descriptive sense, and yet there are a number of algorithms, and peer reviewed authorship which speak purposefully of "rock-finding" algorithms and autonomous science. So what then in this context do we mean by "a rock"? What do we mean by, "not a rock"? And perhaps more importantly, what do we mean by, "a scientifically interesting rock"?

We know that a rock can refer to a wide variety of objects that come in an assortment of shapes, textures, colours sizes and compositions. We know that these wide variations are a result of differences in mineral content and chemical reactions, we know they have been formed by a variety of pressures and temperatures and they may have been altered by still other metamorphic processes which have altered them in some further way. We also know that these rocks now formed are then worn down, weathered by environmental conditions. It might seem then that there is no satisfactory answer to what makes a rock. But then it is in this very variety of compositions and constant processes of remaking which allow for a useful answer to our question.

A rock worth studying, a rock that might be called interesting is called that because of the stories of its formation and reformation. A rock whose structure holds clues unique to the region around it, or those which hold evidence of specific conditions are interesting. To a trained eye many of the processes which have acted upon a rock leave telltale signs. These signs hint at the origin, and subsequent evolution of that particular chunk of the universe. It is then not the form itself which defines it as interesting, but rather the clues to the story of how it came to be. Unfortunately for a visual analysis much of the evidence of this history requires examination at a microscopic or molecular level. There are certain clues, certain geological features that are visible on a larger scale, and thus lend themselves well to an examination at a larger scale.

(a) An example of a visually simple outcrop

(b) An example of a visually complex outcrop

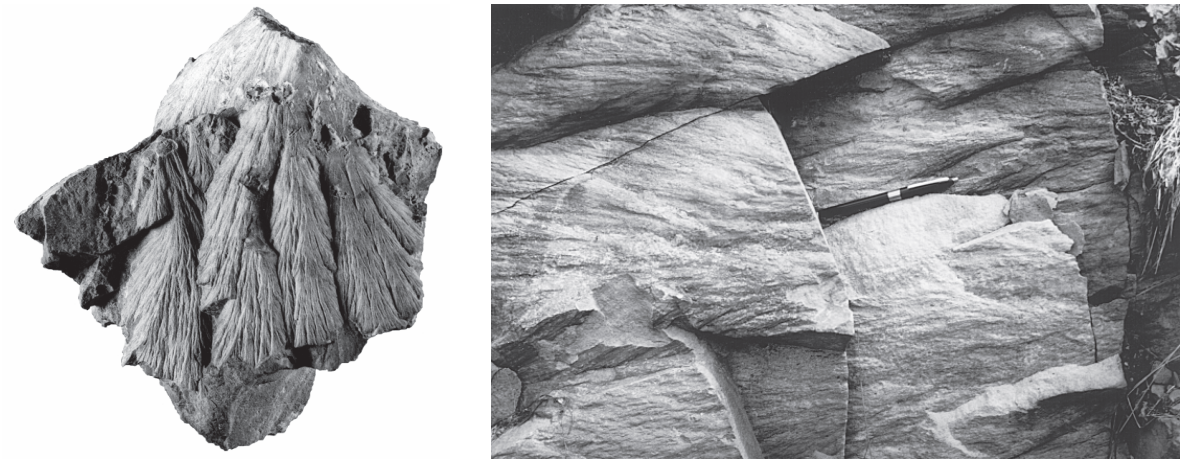
Figure 3.1: While outcrops share some structural components they are quite varied in appearance. This variation is a result of historical conditions which were present in the region

3.1 Outcrops

The first object to discuss are rock outcrops. Outcrops are macroscopic structures where foundational bedrock has been exposed by some process. This exposure of the underlying planetary composition offers the possibility to sample a section of history of the region over a larger geologic time period. They would be of particular interest in the case of a sample return mission because of the clues they may hold to the historical conditions of a region. They are also convenient for obtaining samples since their exposure allows for easy access to normally buried features.

Like the word "rock," the word "outcrop" encompasses a variety of compositions and structures which have arisen from a variety of processes, and thus also have a large variation in colour, size, and shape. Unlike rocks however, the term outcrop is applied only to a specific occurrence of rock structure. Outcrops are prominences that have been exposed through some process that removes the overlaying material most often through processes of erosion, or tectonic activity.

While there can be many different types of erosion, the most vigorous and impressive examples on earth occur through the actions of water (the Grand Canyon for example.) On the majority of examined extraterrestrial bodies, the lack of flowing water means that fluvial erosion plays a much smaller role in the formation of outcrops meaning that the appearance of an outcrop often suggest tectonic or impact events, both of which can be of interest to a planetary geologist. Although it should be noted that some recent discoveries may suggest that there is or was more water on other planets than we originally believed, in which case, the evidence of fluvial erosion on an extraterrestrial body, can be a subject of interest as well.



(a) An example of a nested shatter-cone.

(b) An example of a partially developed shatter-cone

Figure 3.2: Shatter-cones typically will only partially form as seen in 3.2b, due to inhomogeneities in the rocks, but may form nested, "horse-tail" groupings in some types of rocks as in 3.2a

3.2 Shatter-Cones

The second structure we will investigate is a shock metamorphic feature known as a shatter-cone. This is a physical structure which is formed under a relatively small range of pressures, and the formation of which is strongly associated with impact events [31]. To put it in plain language, as far as we know, a meteorite impact is the only natural process by which a shatter-cone will form. Shatter cones are almost entirely unique as a geologic object being the only shock deformation feature which is both distinctive to shock morphology and easily visible to an unaided eye. Ranging in size from a few centimetres to many metres [30, 55], these structures share a number of visual elements that are useful for identifying them, visually.

Shatter-cones are fluted quasi-conical structures, consisting of a curved surface with striations (elongated traces with a preferred direction) which originate from a point along the shatter-cone surface and radiate outwards at small angles, with the striations ranging from 10μm to a few millimetres depending on the grain size of the rock[56]. They form in all types of rocks generally along pre-existing fracture planes, although the level of inhomogeneities within the rock structure may affect the formation of shatter-cones and typically result in partial cone formation aligned to the direction of impact. While typical, the underlying rock type will cause the formation of complete cones, as well as "fractal like" multiple nested "horse tail" structures and multi-directional shatter-cones where the cones are aligned 180 deg apart in direction, next to one another. (See figure 3.3b)

Shatter-cones are created by the shockwaves generated from the deformation the rocks within the crust due to a large extraterrestrial impact. Produced at lower pressures than other shock metamorphic features, shatter-cones can generally be found beyond the "near impact" region were melting and even rock evaporation occurs. Shatter cones are formed by shockwaves with estimated pressures of between 2-20 GigaPascals(GPa) by a process whose precise mechanism is not completely understood. The most unambiguous shatter cones form at pres-

sures greater than 10 GPa while lower pressures 1-5GPa tend to produce features that are less differentiable from features causes by tectonic or volcanic processes.[31] (See figure 3.4 for a comparison of possible metamorphic processes.)

(a) A 30m Shatter-Cone from the Sudbury impact site

(b) Hand Sized Inverted Shatter Cone from the Steinheim impact structure.

Figure 3.3: Shatter Cones can range in size from millimetres to meters and come in a variety of orientations.

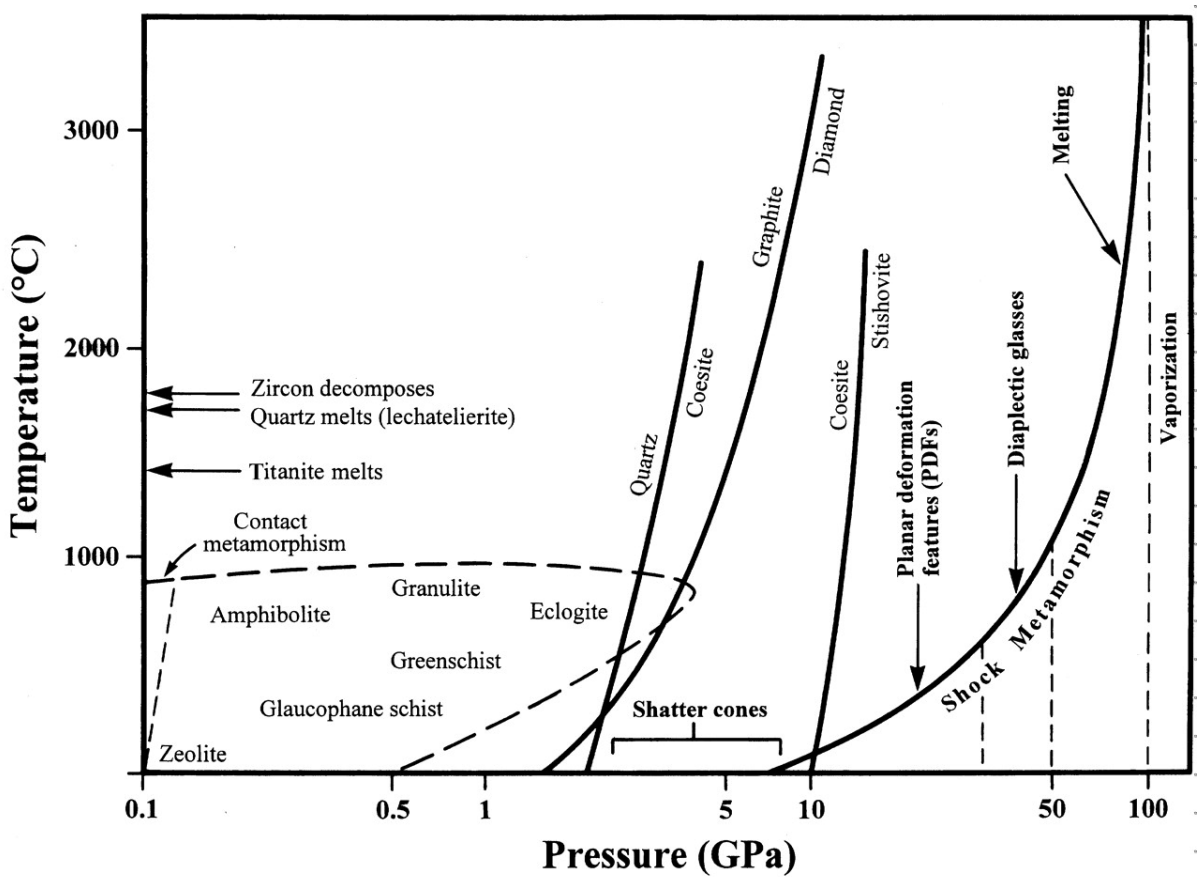


Figure 3.4: This image is a comparison of the conditions for various metamorphic processes in a pressure-temperature plot. In the lower middle of the plot we find the conditions for the formation of shatter-cones.

Chapter 4

Feature Extraction

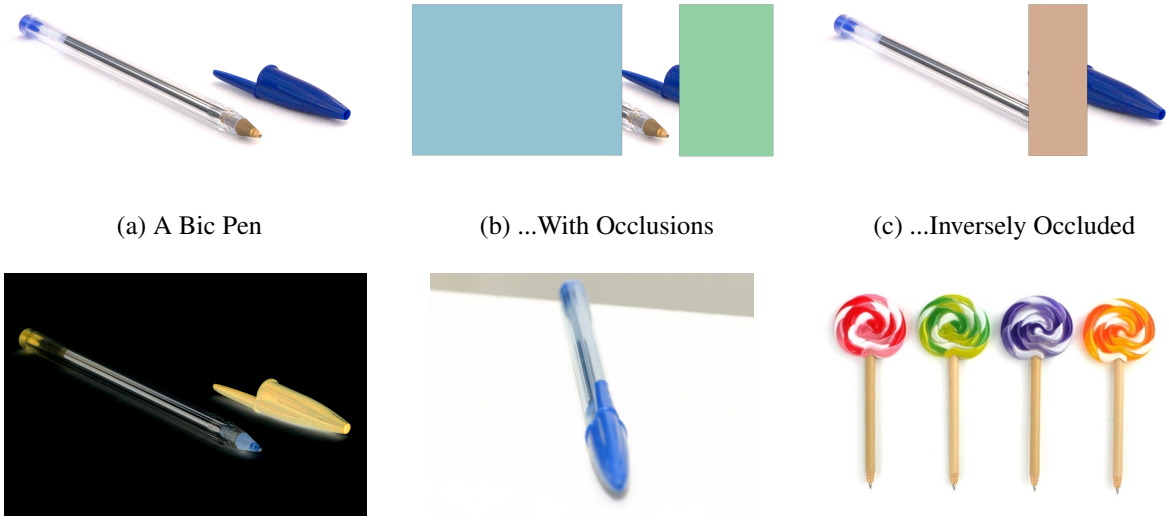
4.1 Teaching a Computer To "See"

For most humans the act of seeing is not something that requires self-understanding, analysis or even conscious thought. To almost everyone seeing is such an intuitive an act that we are rarely aware as to the mechanisms that allow us to see. Imagine a pen sitting on a desk. Now ask yourself: "How do I know that this thing in my visual field is a pen?" What is its quintessential *pen-ness*? Perhaps it is the long thin shape, or the blue colour, or a uniquely recognizable feature such as it's ball point nib. Perhaps it is a combination of features that make it recognizable. All of these suppositions are entirely reasonable assumptions... but are they correct?

If we examine the 6 different images of pens shown in Figure 4.1 we see that despite the wide variation in the configuration and appearance of the various pens, our ability to recognize each object as a pen is almost immediate. Figure 4.1a is the object as a whole, and yet, even when we occlude alternative sections of the pen in figures 4.1b and 4.1c we are still able to identify it as a pen despite no single piece of the image being duplicated. Likewise most people have little difficulty when we change the colour information as in 4.1d or the effect of foreshortening its shape by changing the pen's orientation as in 4.1e. Finally if you look at the pens in figure 4.1f you will find that to identify the object as a pen, most of the visual information must be discarded in favour of a small portion (the ballpoint nibs on the end) in order to identify the object as a pen.

This ability to identify objects is, in and of itself, a difficult challenge if we wish for a computer to "see" an object, and yet even in the previous example, we have simplified a number of details in the way our visual system operates . How do we know where in our visual field the pen is? Is it large and far away? Is it small and close up? What delineates the end of the table and the beginning of the pen? How do we identify the nib, the body and the cap of different pens, even though they may look quite different from one pen to the next. Would you be able to identify the nib of a pen if one were placed in front of you without any context? How is it you know the size of the pen in figure 4.1a when you lack stereoscopic clues, and there are no other visual cues to reference it against?

As you may begin to appreciate, the "simple" act of seeing something is quite far from simple. The human visual system is enormously sophisticated, and seeing is a complex, involved,



(a) A Bic Pen (b) ...With Occlusions (c) ...Inversely Occluded

(d) ...Colour Inverted (e) ...Differently Orientated (f) Lollipop Pens

Figure 4.1: Despite a wide variation in visual information, humans are able to easily identify a pen, despite the fact that much of the visual information might be conflicting as seen in Figure 4.1f

and multi-level process that is the product of a phenomenal amount of neural processing by the human brain. It involves specialized neurological structures which have evolved to process and make sense of a busy and constantly changing visual field. It sets before the field of computer vision a complex model of what must be accomplished if a computer is to *see*. In order to allow a computer to see it must either mimic human processes to a certain degree, or develop computer specific ones that perform similar functions.

The first step in working with computer vision, is to establish a method by which various elements within an image can be described, and identified in a useful way. An image stored on a computer is in many ways analogous to a just seen image on the human eye. In the eye, the stimulation of rods and cones changes the rate of fire of the attached neurons allowing for the differentiation of colour and brightness at each location. In the computer an image is stored as a matrix of values, which represents the intensities of colours at each pixel. While this format is useful for the reproduction and display of an image, it does not provide context for what is contained within the image.

In human vision from the very start, before the electrical signals have even left the retina, the human brain finds ways of organizing data to help change visual information from a series of individual points into meaningful groups of pixels. The receptors in the eyes are bound in excitatory and inhibitory groupings comparing light intensity registered by each receptor to those around them. These groupings allows the eye to identify strong changes in contrast and colour and helps the visual system identify the edges of objects in our visual field. These edges, colours and intensities are then combined to create visual *words* which more complex visual processing mechanisms use to create a visual language our brains can access to identify the objects we see.

It is this ability to create a series of visual words that must be mimicked in order to allow a computer to identify key objects within an image. In Computer Science these words are referred to as image features. These image features range from representations of a single pixel and its immediate neighbourhood, to a description of the image as a whole. They can describe a collection of key points or key regions, or the relationships between those points and regions. Each one will offer a description that can be analyzed, classified, and identified by an algorithm. Each of these words allows for the creation of a bag of visual descriptions for elements within an image.

Below is a brief overview of the features we used in our work.

4.2 SIFT

Scale-invariant feature transform(SIFT) features are a commonly used computer vision feature first published by Professor David Lowe from the University of British Columbia in 1999 [42]. The SIFT feature has become popular in the field of computer vision as a robust and useful feature. Part of the utility of the SIFT feature comes from the fact that it is formally invariant to translation, rotation, and zoom of an image. It has proven to be resistant to noise, blurring, and contrast changes, as well as having utility despite image deformations and affine transformations.

Each SIFT keypoint is a description of a given point on the image and its immediate neighbourhood. Each descriptor depends on 4 key variables: the horizontal position of the neighbourhood centre (x), the vertical position of the neighbourhood centre (y), the keypoint's scale-space (σ), and the neighbourhood's dominant gradient orientation (θ). These variables constitute the SIFT frame which describes the keypoint's location scale and orientation. The feature's descriptor encodes the surrounding neighbourhood gradient information (what Lowe named the *keypoint descriptor*) into a 128-dimensional vector which can then be compared to similarly described points [43].

In his description of his method, Lowe identifies a 4 stage approach to creating the SIFT descriptor:

1. Scale-Space Extrema Detection
2. Accurate Keypoint Localization
3. Orientation Assignment
4. Keypoint Descriptor

This process can be further divided into 8 distinct algorithmic steps as seen in Table 4.1. Steps 1-3 correspond to Lowe's scale-space extrema detection stage. Steps 4-6 correspond to the accurate keypoint localization stage and steps 7 & 8 correspond to the orientation assignment and keypoint descriptor stages respectively.

The first step to Lowe's method is done to make the SIFT descriptor invariant to scale. This is done with a cascade of increasingly blurred images created by convolving the original image $l(x, y)$ with a variable-scale Gaussian filter $G(x, y, \sigma)$ where the scale space $L(x, y, \sigma)$ is described by the function:

$$L(x, y, \sigma) = G(x, y, \sigma) * l(x, y)$$

Stage	Description
1	Compute the Gaussian scale space
2	Compute the Difference of Gaussians
3	Find candidate keypoints
4	Refine candidate keypoints with sub-pixel precision
5	Filter noise unstable keypoints
6	Filter edge unstable keypoints
7	Assign an Orientation to each keypoint
8	Build the keypoint descriptor

Table 4.1: Sift descriptor Algorithm steps. (Adapted from Otero and Delbracio [50].)

This cascade of images is divided into octaves where each octave represent a doubling of the term σ and each octave is itself divided into s intervals such that $k = 2^{1/s}$.with $s+3$ images produced for each octave. For each octave the the sampling distance (δ) is also doubled. The resultant scale-space can be seen in figure 4.2.

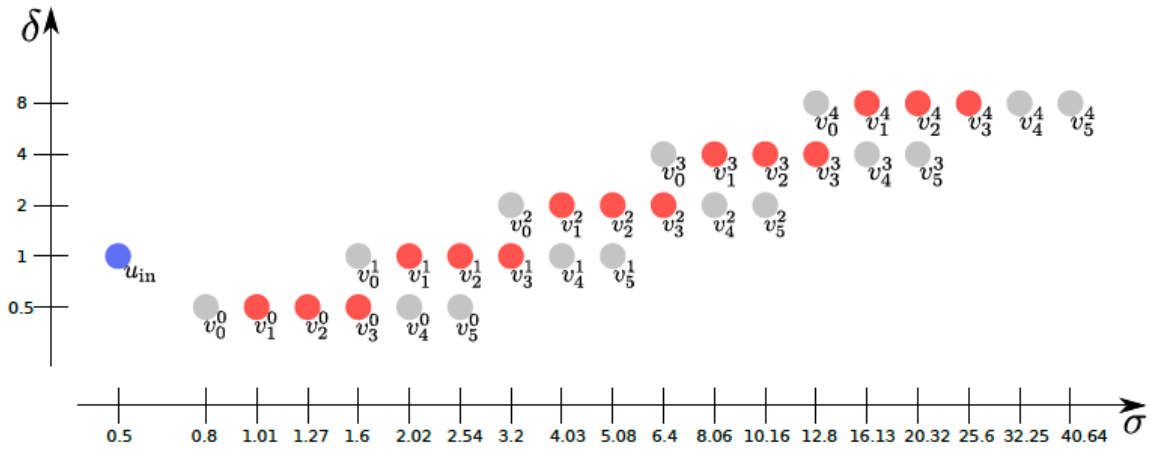


Figure 4.2: A representation of a standard scale space where each dot represents an interval image v_s^o Where o is the image's octave and s is the interval. The dots in red represent the images covering the whole octave while the dots in grey are used in extrema detection.

The second stage is the calculation of the Difference of Gaussian (DoG) scale-space. Lowe suggests that this step is necessary for true scale invariance by approximating the normalized Laplacian $\sigma^2\Delta$ [43]. The Difference of Gaussian scale-space is calculated by subtracting adjacent blurred images in each octave. Thus the DoG image, $D(x, y, \sigma)$ is calculated by:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= G(x, y, k\sigma) * I(x, y) - G(x, y, \sigma) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

The resultant DoG scale-space consists of the same number of octaves with s+2 interval images in each octave as seen in figure 4.3.

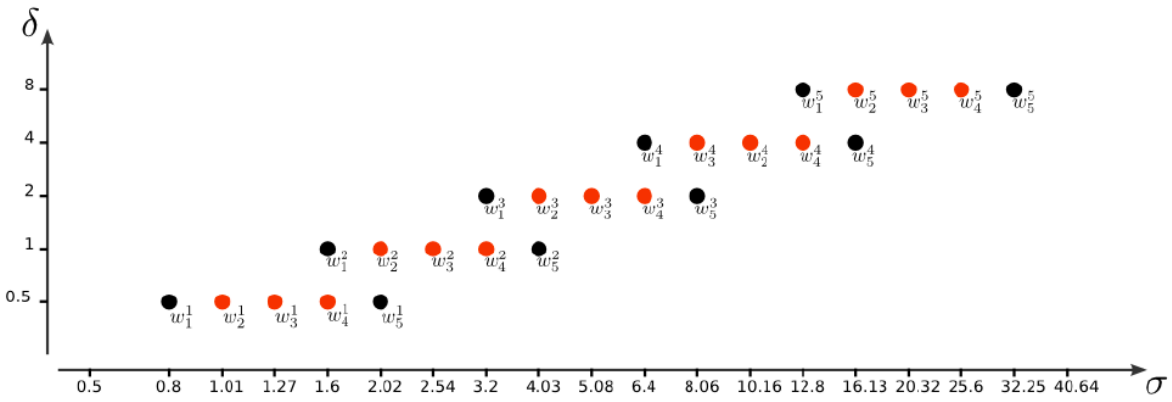


Figure 4.3: A representation of the Difference of Gaussian(DoG) scale space where each dot represents an interval image, w_s^o Where, o is the image's octave and s is the interval. The dots in red represent the images covering the whole octave while the dots in grey supplementary images used for the extraction of candidate key points.

After establishing a DoG scale-space, keypoints are extracted by finding the continuous 3D extrema of the DoG scale-space. Practically speaking this must be done in two steps because of the discrete nature of the DoG scale-space. The first step detects the discrete extrema by comparing each pixel with its 26 neighbours, 8 neighbours within the image and the 9 corresponding pixels in the interval image above and below (see figure 4.4.)

Using a discrete method of detecting extrema on a continuous function causes intrinsic inaccuracy leads to unstable detections, and a susceptibility to noise. While Lowe's original method did not have an interpolation step, Mathew Brown suggested a method of obtaining sub-pixel accuracy using a second order Taylor expansion to approximate the underlying function[5].

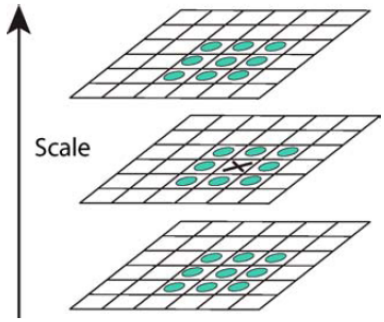


Figure 4.4: Extracting candidate keypoints by finding 3D Extrema.

The final step in determining accurate keypoint localization is to reject inherently unstable keypoints. Two main classes of keypoints are rejected at this point. Keypoints in areas of low contrast being inherently unstable, are eliminated using a threshold and keypoints along edges which may be translation invariant along that edge are eliminated using a Hessian Matrix.

The SIFT descriptor achieves its rotation invariance because it aligns the keypoint descriptor along the dominant gradient angle of the nearest Gaussian smoothed image. Once the

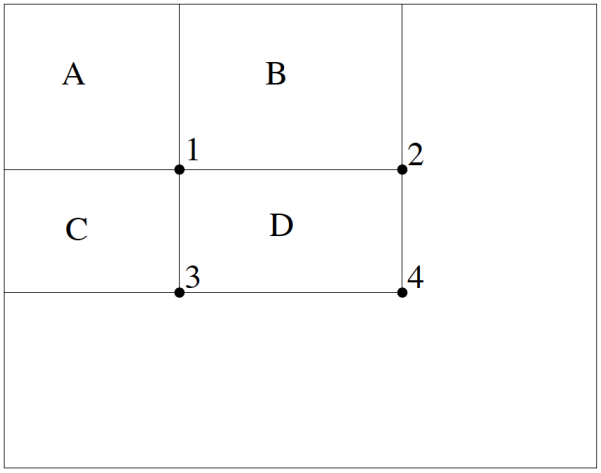


Figure 4.5: The sum of the pixels within rectangle $D_{(1,2,3,4)}$ of the original image can be computed using the four corner points within the integral image. Point 1 gives the area of A, Point 2 the area of A+B, Point 3 the area of A+C, Point 4 the area of A+B+C+D. This means we can find the value for area D. ($D = 1 + 4 - (2 + 3)$)

dominant gradient angle θ is found an keypoint descriptor is generated by using a 16x16 matrix is normalized about θ. The dominant gradient of each element of this matrix is assigned to one of 8 orientations which are grouped in 4x4 regions. This generates a 4x4x8 matrix which forms a 128 dimension description vector for the keypoint.

4.3 Haar Like Features

Haar like features are a name given to an image feature first described by Paul Viola and Michael Jones in their 2001 paper, *Rapid Object Detection using a Boosted Cascade of Simple Features* [63]. Haar like features are an intensity based feature that utilize an additive method for comparing adjacent rectangles. Viola and Jones' method also adds a cascade of features, which helps reduce the computational complexity while maximizing accuracy. The result of this combination produced a comparatively accurate detection algorithm which was much more efficient than previous work (Viola and Jones claimed their algorithm was 15 times faster than previous work.)

The Viola/Jones implementation of Haar like features begins with the creation of an image representation the authors called an *integral image.* The integral image is a 2D vector where the value at (x, y) in the integral image (ii) is a sum of all of the pixel intensities in the image to be processed to the left and above the same point on the original image $i(x', y')$.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

The integral image is created in order to allow for the rapid summation of image intensities. In figure 4.5 the sum of image intensities within a given rectangle (ii_1, ii_2, ii_3, ii_4) can be calculated using the values found at the four corners of that rectangle. Since the value of a point is

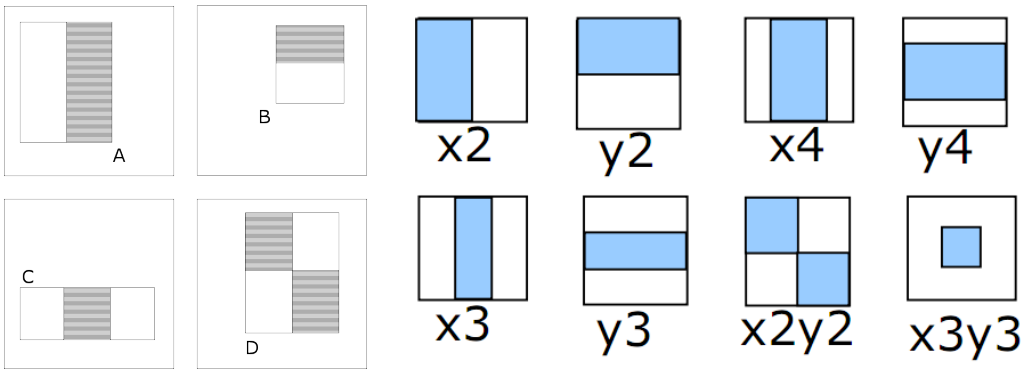
the sum of the intensities of the area above and left, we see that:

$$\begin{aligned} \text{the point} &= \text{is the sum of intensities of the area represented by:} \\ ii_1 &= A \\ ii_2 &= A + B \\ ii_3 &= A + C \\ ii_4 &= A + B + C + D \end{aligned}$$

From this we can see that the area of D is equal to:

$$D = (ii_1 + ii_4) - (ii_2 + ii_3)$$

This ability to quickly compare the intensity of image areas allows for the creation of computationally cheap features. Viola and Jones used the differences in intensities between adjacent squares within a 24x24 pixel subsection of an image in order to create weak classifiers (see figure 4.6a.)



(a) An example of features used by Viola and Jones for face detection

(b) An extended feature set

Figure 4.6: Haar like features created by difference combinations of additive and subtractive non-overlapping intensity area summations.

4.4 EOG

Edge Orientation Gradients are one of the oldest histogram features and were first described by William T. Freeman and Michal Roth in their 1994 paper, *Orientation Histograms for Hand Gesture Recognition* [29]. Freeman and Roth found that localized pixel windows could be used to determine the dominant gradients within a region of an image. In their work they used these regions to produce a descriptor of hand positions. They found that localized gradients were far more invariant to lighting and contrast changes than the larger image descriptors of the time.

Freeman and Roth used derivative operators dx and dy to generate their gradients with $\arctan(dx, dy)$ corresponding to the gradient direction and $\sqrt{dx^2 + dy^2}$ being the gradient contrast. They divided the gradients into one of 36 bins and then blurred and normalized the results to create a descriptor.

In modern algorithmic implementations of EOGs a variety of 2D filters may be used such as sobel operators to determine the gradient in a number of different directions. The histogram bins have been simplified to 5 traditionally consisting of 4 directional gradients (N-S, E-W, NE-SW, NW-SE) and one unidirectional gradient (see figure 4.7.) To make these histograms a useful image feature, many implementations will calculate these localized gradients over a variety of image window scales, to allow for a larger vocabulary of words.

-1	-2	-1	-1	0	1	2	2	-1	-1	2	2	-1	0	1
0	0	0	-2	0	2	2	-1	-1	-1	-1	-2	0	0	0
1	2	1	-1	0	1	-1	-1	-1	-1	-1	-1	1	0	-1

(a) Horizontal (b) Vertical (c) NW-SE Diagonal (d) NE-SW Diagonal (e) Non-Directional

Figure 4.7: 4 Directional Sobel operators and one non directional one too.

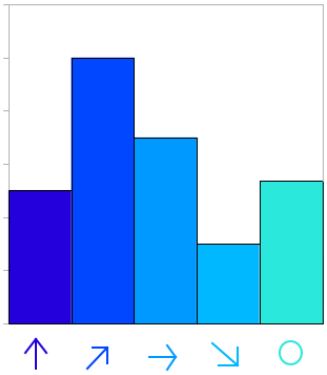
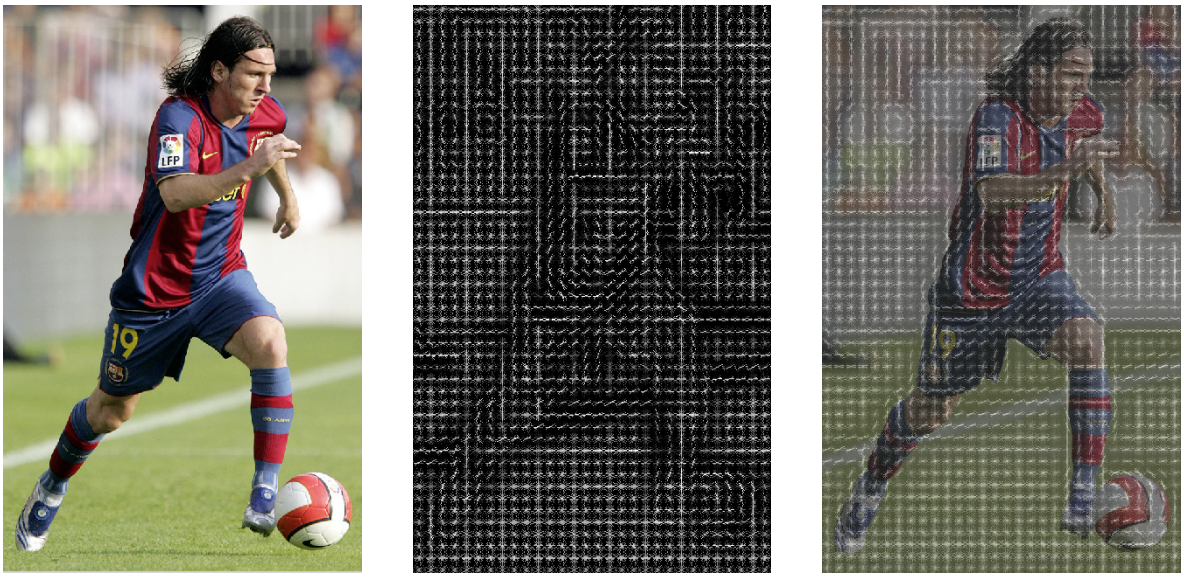


Figure 4.8: A modern Edge Orientation Gradient

4.5 HOG

Histogram of Oriented Gradients (HOG) was developed as a method of edge detection for the purpose of detecting people in images. First devised in 2005 by Navneet Dalal and Bill Triggs, HOG is similar to SIFT, but rather than focusing on key points, HOG is computed in dense grids at a single scale without assigning an orientation[19]. Because it is calculated in a grid, the HOG feature takes advantage of overlapping sub windows to normalize the local contrast (unlike EOG.)

Dalal and Triggs, theorized that the distribution of these local intensity gradients could be used to characterize structures within the image (in their case structures of the human body.) To the human eye a visualization of the HOG features gives enough information to recognize the underlying image as seen in figure 4.9.



(a) An image of a sports player (b) Visualization of HOG features (c) HOG Features on image

Figure 4.9: Histogram of Oriented Gradients were first used to describe the positions of a body for the purpose of identifying that it was a person. Image 4.9c above shows the corresponding gradients from the HOG results in 4.9b of the sports player image in 4.9a

Dalal and Triggs outlined a 6 step process for their use of HOGs (See Figure 4.10.) The Process begins by normalizing the gamma/colour space, Dalal and Triggs reported that they were able to use this process on both grayscale and colour images with the colour information providing slightly better results overall. The second step is it to calculate the gradient of the image, using a gradient filter. In the case of colour images, each colour channel is calculated separately. Dalal and Triggs list four variations of gradient filters they tried, Sobel operators, (as seen in fig.4.7) in addition to three 1-D derivative filters: uncentered $[-1, 1]$, centred $[-1, 0, 1]$, and cubic corrected $[-1, -8, 0, 8, 1]$. Dalal and Triggs reported that the best results came from the 1-D centred filter and its transpose, but only marginally so and while they speculate on reasons, it is not clear that this is universally true.

In the third step, Dalal and Triggs divide the image up into blocks of cells, each cell containing a similar number of pixels, each of whom *votes* for an orientation bin based on the gradient magnitude at its location. The orientation bins are evenly spaced over the whole rotational range ($0° - 180°$ for *unsigned* gradients and $0° - 360°$ for *signed* gradients.) These votes are then interpolated bilinearly between neighbouring orientation and position bins. The cells are made up of a block of pixels arranged spacially in either rectangular or a radial orientation. Each of these cells is then assigned to a series of overlapping blocks such that each cell belongs to a number of adjacent blocks. (i.e. a stride of $\frac{1}{2}$ of the block size ensures that every non edge cell belonged to 4 adjacent blocks. This overlapping nature is imperative for the Dalal and Triggs next step.

The fourth step of the process is contrast normalization. Because lighting conditions change fairly drastically in an image, the overall contrast variation in any given local sub-window can be wildly different from surrounding sub-windows. Because Dalal and Triggs overlap their

image blocks, they ensure that each region is normalized with a number of surrounding blocks. Dalal and Triggs explored four different methods for block normalization.

Let h be the non-normalized vector of histograms in a given block.

$$\|h\|_k \text{ is its k-norm for } k = \{1, 2\}$$
$$\text{and } e = \text{ some small constant.}$$

The normalization factors tried were the following:

$$\text{L2-norm:} \quad f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}}$$
$$\text{L2-hys:} \quad \text{L2-norm with } v_{max} \text{ limited to 0.2 and renormalized.}$$
$$\text{L1-norm:} \quad f = \frac{v}{(\|v\|_1 + e)}$$
$$\text{L1-sqrt:} \quad f = \sqrt{\frac{v}{(\|v\|_1 + e)}}$$

In their experiments, Dalal and Triggs found the L2-Hys, L2-norm, and L1-sqrt schemes provide similar performance, while the L1-norm provides slightly less reliable performance; however, all four methods showed very significant improvement over the non-normalized data.

After performing this normalization, the feature vector is created by collecting each cell from each block of the detection window to create a feature vector.

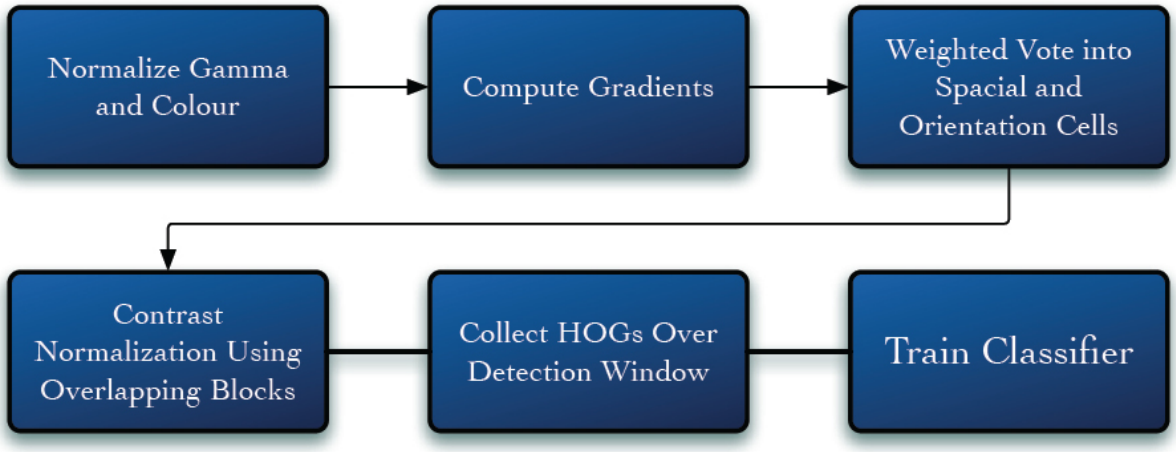
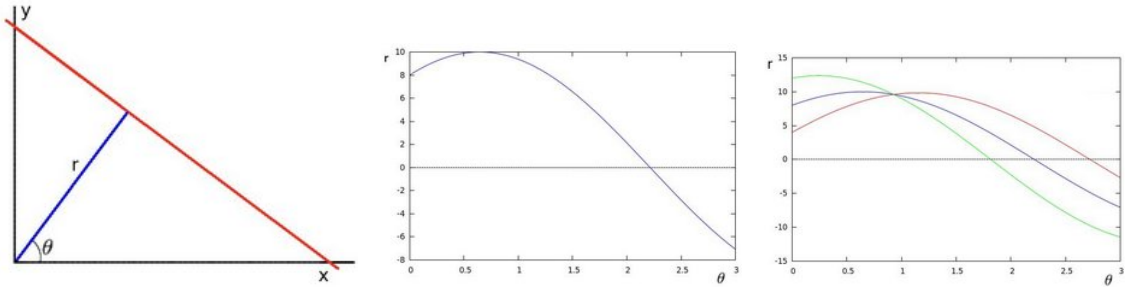


Figure 4.10: Dalal and Triggs described a six step process in their 2005 paper.[19]

4.6 Hough Transforms

The Hough transform is a technique used to isolate features of a particular shape within an image. The hough transform is generally implemented in two flavours, one which can be



(a) Rho Theta Notation (b) A Point described in Polar Space (c) Points on a line

Figure 4.11: In 4.11a a point in space can be described by it's distance from, and rotation about the origin. 4.11b is that point as described in Polar Space. 4.11c shows the polar representation along a straight line in Cartesian Space.

parametrized to find specific shapes, and a more general one that is more commonly used as an image descriptor. Since the first incarnation of this algorithm requires a paramiterization of the desired shape, it is generally reserved for, regular curves such as lines, circles, ellipses etc. One of the key strengths of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise.

The Hough Transform takes its name from Paul Hough, who rst described the fundamental principles in a 1960 patent for his work in bubble chambers at the U.S. Atomic Energy Commission. [7]. Hough's methods were improved upon and the algorithm optimized for image detection by Duda and Hart in their 1971 paper, *Use of the Hough Transformation to Detect Lines and Curves in Pictures* [23]. One of the key differences in the work of Duda and Hart, is their use of rho-theta notation. Traditionally image notation is given in Cartesian Space, thus a pixel will be denoted as the combination of the horizontal and vertical offset from the origin. (x, y) and a line can be denoted as a function of a series of pixels $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ if the line is a regular curve, it can be described using two parameters, $\{m, b\}$ where m is the slope and b is an offset. Thus any regular curve can be written as:

$$y = mx + b \tag{4.1}$$

The problem with the use of Cartesian Space was that m could be infinity. Duda and Hart's use of rho-theta notation prevents this from happening. Duda and Hart translated point descriptions from Cartesian Space to Polar Space thus preventing $m = \infty$. In figure 4.11a we can see that a pixel in Cartesian Space can also be described as a function of its distance from, and rotation about, the origin. Thus we can write it as:

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right) \tag{4.2}$$

This can then be arranged into polar coordinates:

$$r = x\cos\theta + y\sin\theta \tag{4.3}$$

this means that for any given pixel (x_1, y_1) a representation in polar space can be given as a function of all of the lines passing through that point as:(see figure 4.11b for the curve produced by $x_1 = 8$ and $y_1 = 6$)

$$r = x_1\cos\theta + y_1\sin\theta \tag{4.4}$$

If the same operation is performed for all the points in an image it will produce a series of sinusoidal curves in polar space. If the curves of two different points intersect in the plane $\theta - r$, we can say those points are along the same cartesian line. For instance, following expanding on the example above if by plotting two more points along a line: ($x_2 = 9, y_2 = 4$ and $x_3 = 12, y_3 = 3$, we get the result shown in figure 4.11c.

We can see from equation (4.2) and equation (4.1) that this intersection is comes from the parameters m and b in cartesian space equate to a function of r and θ in polar space.

$$m = \left(-\frac{\cos\theta}{\sin\theta}\right) \tag{4.5}$$

and

$$b = \left(\frac{r}{\sin\theta}\right) \tag{4.6}$$

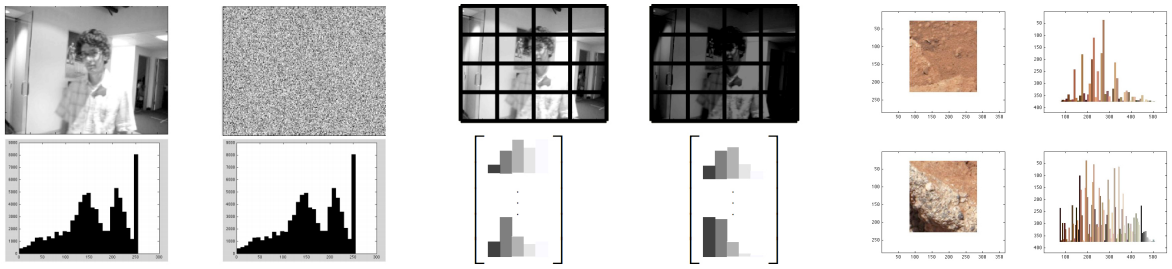
The fact that the curves of points on a line intersect allows for powerful tools of analysis. By assigning a value to each pixel along the curves in Polar Space, a cumulative energy value can be assigned to points and an energy map produced, whereby those points in r, θ space with the highest values correspond to the most dominant linear feature. Likewise lines which are not quite linear but fall near a line can be grouped by counting all values within a given neighbourhood in Polar Space. Likewise since parallel lines in cartesian space have the same slope(m) equation (4.5) suggests that parallel lines will be located at peaks same point along the θ axis in polar space.

Since the Hough Transform is a translation between Cartesian and Polar Spaces, performing a translation on an image would result in a curve for every pixel, rendering the results pretty much meaningless. In order to maximize the value of the Hough transform, it is beneficial to maximize the information contained in the pixels entered into the transform. Since edges generally contain the largest amount of information, an edge detection algorithm (such as the canny edge detector) is generally used to produce a binary value image,which is then passed into the hough transform.

To use a Hough Transform as a general image descriptor for a machine learning program, hough peaks are calculated and labeled biased on a threshold value. Those peaks are then clustered either in two dimensions or one (keeping in mind that (4.5) suggest that clustering along the θ access will give you an indication of parallel lines.) These clusters are then used to create an image descriptor.

4.7 Intensity and Colour Histograms

Compared to other image features, intensity and colour histograms are rather simple. Simply put, an intensity histogram is a count of the intensity values of the pixels within a given window or in the case of colour histograms the intensity values of each channel over a given window. The intensity value is assigned to one of a series of equally spaced bins, and the total number of pixels assigned to each bin is returned as the image descriptor. Intensity histograms capture no spacial information and so vastly different images may have similar histograms (see figure 4.12a.) In order to minimize this effect, it is often helpful to use smaller image sub-windows in order to preserve some of the spacial information in the image. Intensity histograms are also



(a) Identical Intensity Histograms for dissimilar images

(b) Susceptibility of Intensity Histograms to Lighting Changes

(c) Rock VS. Regolith

Figure 4.12: With out careful application, intensity histograms can exhibit a number of different weaknesses. Properly applied however they form a useful tool.

very susceptible to lighting changes, and so the same image taken with different lighting may create vastly different histograms as well (see figure 4.12b.)

Despite these obvious weaknesses intensity histograms can still be useful within their intrinsic limitations. Histograms offer a computationally low cost method of describing an image patch, and as can be seen in figure 4.12c, they seem to be useful in the case of our work in particular because of the relatively homogenous colouring of large parts of extraterrestrial regolith. This same attribute also offers promise in its ability to help differentiate between different types of rocks as well.

4.8 Edge Density

Edge Density is an image descriptor which as its name implies calculates the density of edge pixels located within an image or image sub-window. The edge density is found by using a binary edge detection algorithm such as a canny edge detector and then the density of the edges is calculated using the formula:

$$f = \frac{1}{a_r} \sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} e(x, y) \tag{4.7}$$

where:

$$a_r = (x_2 - x_1 + 1)(y_2 - y_1 + 1) \tag{4.8}$$

Where $e(x, y)$ is the detected edge image of a sub-window $i(x, y)$ of the original image i. a_r is the area of the image patch and (x_1, y_1) is the top left, and (x_2, y_2) is the bottom right pixel.

Chapter 5

Machine Learning Algorithms

Once image features have been extracted from an image, there must be a way of giving those features meaning. Image features are in essence a group of words which can be used to describing what an image contains, but as yet we do not have the vocabulary to describe the contents of an image. For a human being, the development of linguistic context makes up a large part of our early years. Through trial and error and experimentation a child learns to classify the things they see into meaningful groupings based on contextual clues. The contextual and structural rules the child develops are associated with the signals arriving from sensory organs, and through repeated experience allows for a system of comparison which allows a child to classify novel objects from similarities to known things.

This process of learning through known examples is referred to as "concept learning" in psychology [6]. In computer science this is known as "supervised learning" because an outside expert is required to "supervise" the learning process by providing a training set for the computer to learn on. It is this process of supervised learning that is used to create a language to describe features within images.

5.1 What is Machine Learning?

In the most general sense, machine learning can be thought of as a computer learning from past experience, or in the words of Tom Mitchell,"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E[45]."

Machine learning differs from what might traditionally be thought of as programming, because the computer is not given explicitly coded instructions of how to accomplish a task, rather it is given a set of rules by which it can evaluate its performance(P in Tom's words) and a way of revising it's method of performing a task (T). There are a wide variety of learning strategies, but they generally fall into three main categories.

Supervised Learning Supervised learning, as we alluded to above, is when an algorithm is given a set of objects which have previously been identified, and are given to a learning algorithm to evaluate. Once the algorithm has classified the images, the results are evaluated and some form of refinement is applied, either from a weighting function based on the identified,

"ground truth," or from human input. This is then repeated until an acceptable accuracy is achieved or no further improvement is gained.

Unsupervised Learning Unsupervised learning is a process in which no a priori classification is given, but rather, an algorithm uses some global weighting function to evaluate its performance at a task. In contrast to supervised learning which attempts to differentiate a set of data into a known structure, an unsupervised learner tries to identify an underlying structure that best describes the data it has.

Reinforcement Learning Reinforcement learning in which a computer tries to maximize a cumulative "reward" function over time, predicated on the value given for the correct outcome being achieved. Unlike supervised learning, the algorithm is not given pre-evaluated data, but unlike unsupervised learning the classification of the data is known. The algorithm is simply rewarded for more optimal results.

Of these strategies of learning, we will focus on supervised learning since we are looking for a specific class of objects and we are able to relatively easily provide labeled images. More specifically we will utilize an algorithm known as AdaBoost as we will discuss in section 5.2

The idea of leaving a computer on its own to figure out how to classify objects, might seem counter intuitive at first. A human is generally far more adept at classifying objects than a computer, especially in the visual realm. It seems logical to hypothesize that a human could devise a more accurate methodology for separating description vectors than could be done algorithmically. So why not simply tell a computer what to look for?

There are two strong arguments to be made against such a strategy. First, humans are often able to do things instinctually, without consciously understanding the cognitive methodology used to arrive at a given solution. For an example, would you be able to describe the process by which you recognize your grandmother's face? Even if you were able to document the process, how would those steps translate into a heuristic a computer might use? To go farther some have argued, that there are perceptions that are literally indescribable. These perceptions are called *qualia* by philosophers, and they represent a concept so familiar to a human's understanding that it cannot be described to someone who lacks that perception [20]. How do you describe the colour you see when light with 650 nm wavelength is directed at you?

The second reason is that structural differences in the computational hardware of humans and computers differs such that a computer and a human have different optimal strategies. Computers have massive capacities for instantaneous memory recall and an extraordinarily rapid mathematical processing speed, while human brains have structures that operate in parallel, and are exceedingly good at comparative analysis. A system that seems intuitive and simple to a human may rely on complex underpinnings when attempted by a computer.

5.2 AdaBoost

Hypothesis boosting, or "boosting" is a machine learning technique that uses an iterative process to generate a more accurate classification heuristic from a number of "weak hypotheses."

Stemming in large part from theoretical work by Leslie Valiant who first described a methodology of autonomous learning which became known as Probably Approximately Correct (PAC) theory[62], by using a number of these moderately accurate PAC learning strategies, boosting is able to create a heuristic of arbitrarily high accuracy[57].

First described by Yoav Freund and Robert E. Schapire, in 1995, Adaptive Boosting, or AdaBoost [32] is a derivation of boosting in which unlike prior algorithms, adjusts adaptively to the errors of the PAC hypotheses. Beginning with equal weighting for all examples, incorrectly classified examples are given more weight after each iteration, increasing their value which forces the algorithm to give more weight to marginal or difficult cases.

5.2.1 The AdaBoost algorithm

The AdaBoost algorithm operates over a set of known training examples S. Each of these examples is comprised of two parts. The first is the domain or instance space (N) which is the descriptor given to the example. (In the example in figure 5.1, the instance space would be the x and y co-ordinates.) The second part is comprised of a finite label space (L) which correspond the different classification groups of the objects to be learned. In figure 5.1, the label space is represented by the + and − signs and $L = \{+1, -1\}$ to begin with. Our training set of m training examples will consist of m objects, each with one instance descriptor, n_i where $n_i \in N$, and one label descriptor, l_i where $l_i \in L$ and $l_i = \{+1, -1\}$.

In addition to the training set, the AdaBoost algorithm requires a *base* learning algorithm. This learning algorithm, must take as input the training examples, S and a distribution of weights, D over the the training examples $\{1, \ldots, m\}$.

$$S = \langle (n_1, l_1) \ldots (n_m.l_m) \rangle; \qquad n_i \in N, l_i \in \{-1, +1\}$$

$$\sum_{i=1}^{m} D(i) = 1$$

where the initial weighting of the hypothesis is the same across the distribution:

$$D(1) = \frac{1}{m}$$

From this the learner computes a *base hypothesis* (h). This hypothesis takes the form of a real value in the instance space,($h : N \to \mathbb{R}$) where for each example in S the sign of $h(n)$ is the label predicted for each n and $|h(n)|$ is the confidence in the prediction, where the farther $h(n)$ is from 0, the greater the confidence of the prediction. The weak hypothesis $h(n)$ is calculated to minimize the error function (ε) across the weighted distribution D.

$$\varepsilon = \sum_{i=1}^{m} D(i) I(l_i \neq h(n_i))$$

(I is an indicator function)

The AdaBoost algorithm will then iterate T times over the training set to arrive at a final hypothesis which is a discriminant function H over the sum of weighted weak hypotheses ($\alpha_i h_i$) of each iteration t, thus:

$$H(n) = \sum_{t=1}^{T} \alpha_t h_t(n)$$

Where $\alpha_t \in \mathbb{R}$ and is calculated based on ε_t, typically (as seen in the example in 5.1):

$$a_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t}$$

The main advantage of AdaBoost, (and the reason it is named adaptive boosting) is that the distribution D is recalculated after each iteration. This allows for multiple weak hypotheses to be calculated across different distributions, with an emphasis on incorrectly identified training examples.

$$D_{t+1}(i) = \frac{D_t(i)\mathsf{exp}(-\alpha_t l_i h_t(n_i))}{Z_t}$$

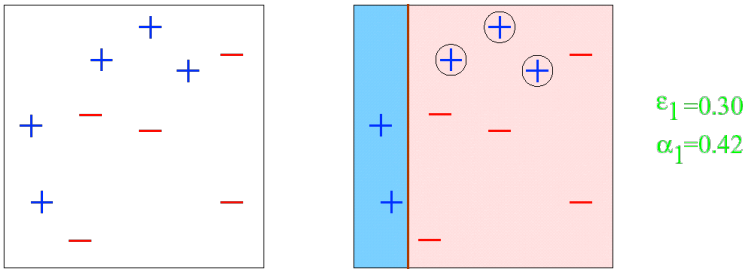
(Z_t normalizes D_{t+1} to a distribution.)

This process ultimately results in a final classifier which is the sign of our final hypothesis $H_{final}(n) = \mathsf{sign}[H(n)]$ as seen in figure 5.1g.

5.3 Training Errors and Cross-Validation

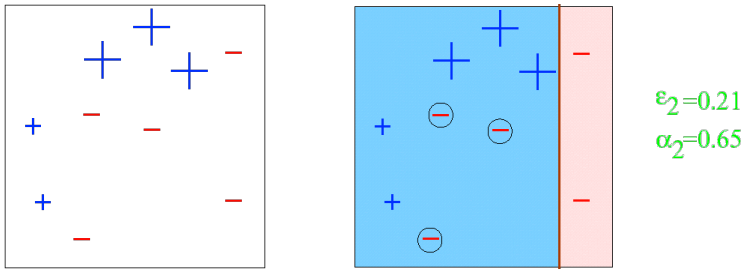
While machine learning algorithms have become popular and powerful tools in the field of Computer Vision and pattern recognition, they are not without potential pitfalls. Ultimately, the goal of these algorithms is to develop a model which can then be generalized so that when a novel object of the same type is encountered it might be identified. All supervised learning algorithms like AdaBoost take a set of labeled training data and use that to devise a model. Since our training set consists of a finite set of objects, we run the risk of developing a model that has errors introduced by the model we choose and the composition of our training set. Within this model creation it is imperative to balance two factors, *bias*, and *variance*. These two factors, collectively known as generalization errors, come from the almost infinite variability of the real world and the finite size of our data set, and data description.

Bias errors arrive from those examples which fall outside the confines of the model do to the limitations in the specificity of the model. These examples would be outside of our model parameters even given very large training sets. They are those data which are not differentiated by the model used. Variance errors on the other hand are those data points which are caused by these poorly differentiated data points when they are included in the training set. This



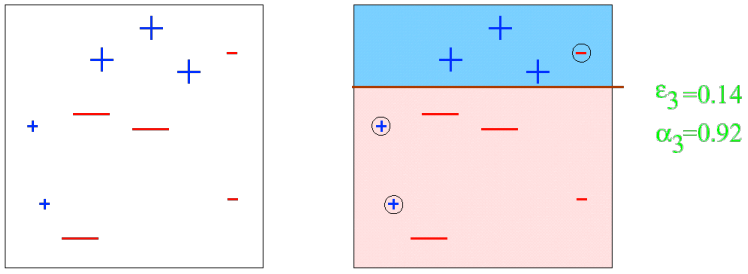
(a) initial distribution D_1

(b) initial hypothesis $h_1(n)$



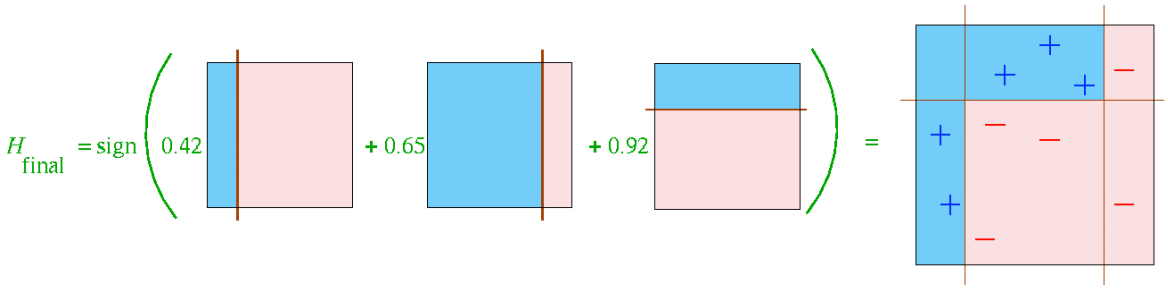
(c) distribution D_2

(d) second hypothesis $h_2(n)$



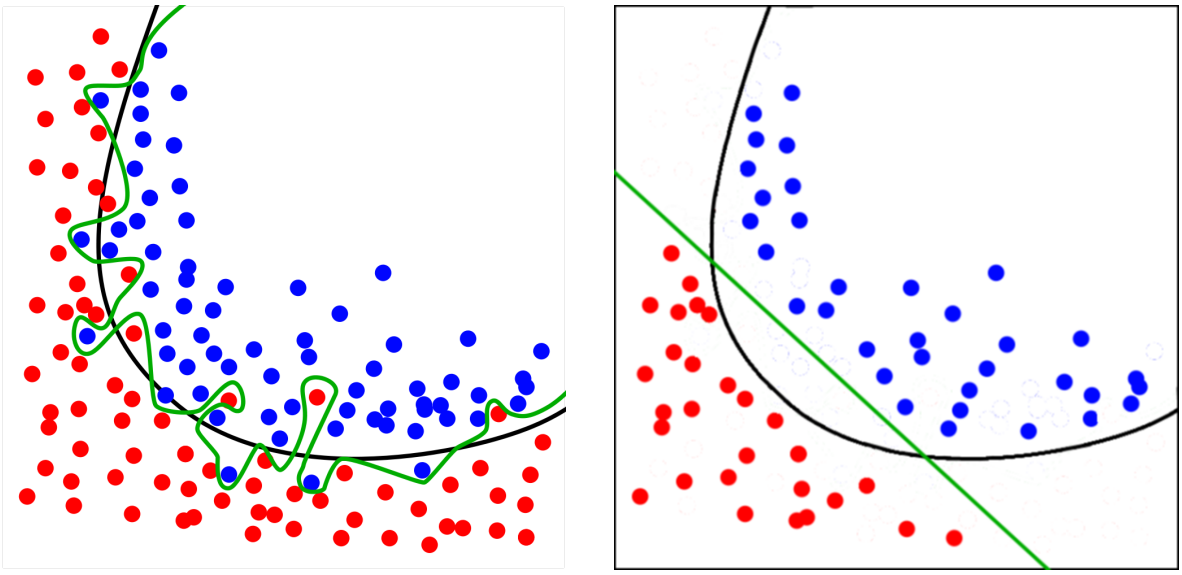
(e) distribution D_3

(f) third hypothesis $h_3(n)$



(g) Final Hypothesis H_{final}

Figure 5.1: A simple example of the AdaBoost algorithm. In 5.1a we have an equal distribution weight $D(i) = \frac{1}{m}$ represented by equally sized + and − symbols. 5.1b shows an initial hypothesis separating the blue area and pink area. The circled symbols represent misclassified examples. In 5.1c we see the updated distribution weights with larger symbols representing a larger weighting in the distribution, and smaller symbols smaller weights. 5.1d represents an updated weak hypothesis based on D_2 5.1e and 5.1f are a third iteration of the algorithm. 5.1g shows the final hypothesis $H(n)$ represented by a cumulative sum of the weighted weak hypotheses.



(a) Overfitting (b) Underfitting

Figure 5.2: A statistical model which is overfit to a specific data set represented by the green line in 5.2a while the green line in 5.2b shows a model which is underfit due to an insufficient data set.

inclusion can cause the model parameters to be shifted during training. Generally speaking no sufficiently complex object can be perfectly separated without some of these outlying data points. (Even the human visual system is prone to misidentification, which we often refer to as "optical illusions")

Variance and bias errors tend to negatively correlate to each other for a given complexity of the data model used. The simpler the data model the more influential bias errors will be, while the more complex the model the greater the effect of variance errors. Since by definition, AdaBoost adds complexity for each iteration, we must use enough iterations to minimize the bias of our model while not having too complex a model such that its variance makes it too specific to our training set. One strategy to prevent this from happening, particularly in smaller data sets, is to use a process called *cross-validation*. Cross-validation is a process in which the data we use is divided up into some number of folds (n). Each of these folds is a discrete subset of our training dataset, D where $D = \{D_{f1}, D_{f2}, \ldots, D_{fn}\}$. We then run our training algorithm n times using D_{fn} as a test set and the remaining subsets as the training set. This allows for the minimization of the influence of any given data point on the overall model as the average of each fold is calculated and averaged. This allows for a more complex model to minimize bias errors while mitigating the arising of variance errors.

Poorly trained models can be described as either overfit, or underfit. The *overfitting* of our training data means that our model is too complex and has high variance. It will have problems when applied to the general population. An example of an overfit model is represented by the green line in figure 5.2a, while a good *generalizable* statistical model is represented by the black line. By contrast, too simple of a model runs the risk of *underfitting*. An underfit model will have problems because the model bias means it does not generalize to the distribution of

real world examples.

5.4 Classification and Regression Trees

5.4.1 What are Classification and Regression Trees?

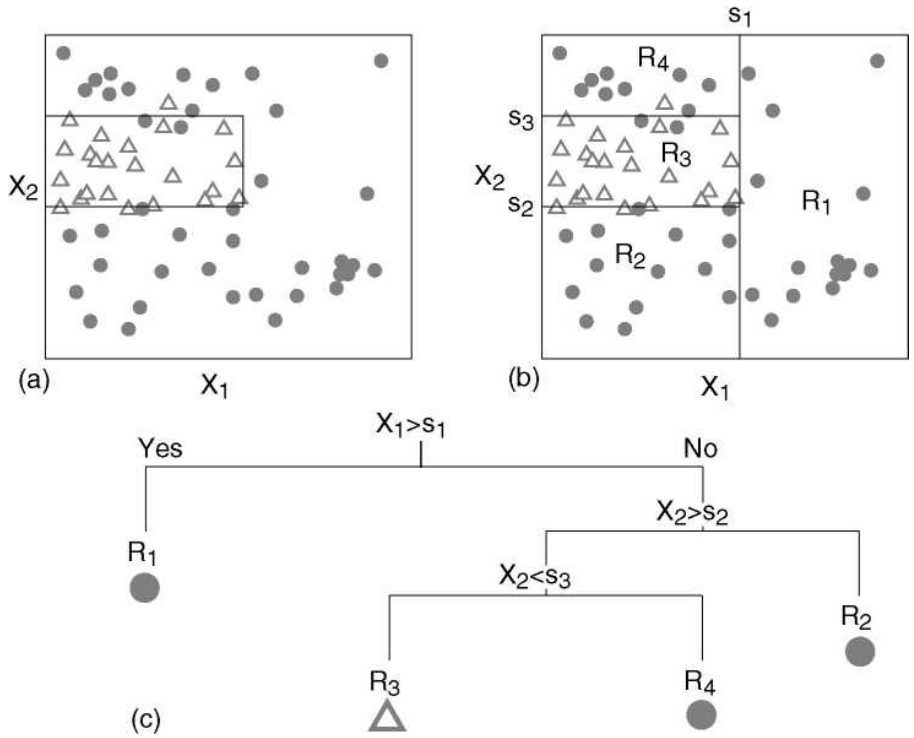


Figure 5.3: The use of Classification and Regression Trees (CART) allow for more complex separations in data spaces as seen in (a). The tree does this by using a series of recursive binary divisions to simplify the data description in each subspace (b)&(c). CART offer a low cost way to create more useful simple classifiers for an AdaBoost Algorithm.

The term Classification and Regression Trees (CART) is an umbrella term, coined by Leo Breiman in 1984 to describe two associated classes of prediction trees used in data mining. While Classification and Regression trees are similar to each other in form, there usage differs in that Classification trees, as the name suggests, are used to predict the general class of an object, while regression trees are used to generate a real number outcome.

The use of CART in predictive modelling offers a number of advantages; it is fast, it offers a low cost way to incorporate more complex divisions in a data space, and in contrast to strategies like linear regression which require a single predictive formula over the entire data space, CART allows for a non-smooth regression space. CART do this through a series of recursive steps. Each step in this partitioning subdivides the data space into subsections which can be described more accurately by a simple classification model.

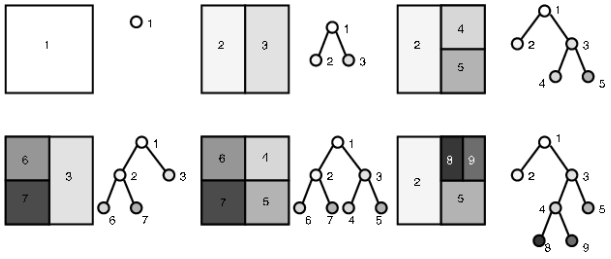
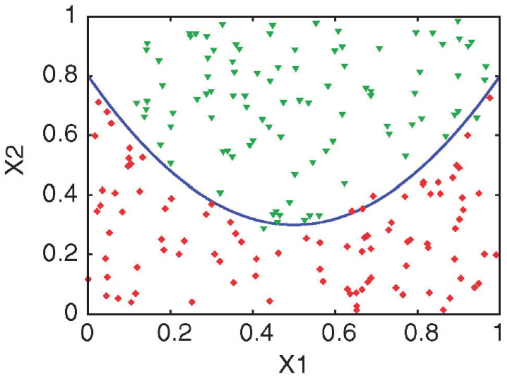


Figure 5.4: The shape of the decision tree used in partitioning affects the division the data space. The Square in the centre of the top row represents a simple binary classifier or stump.

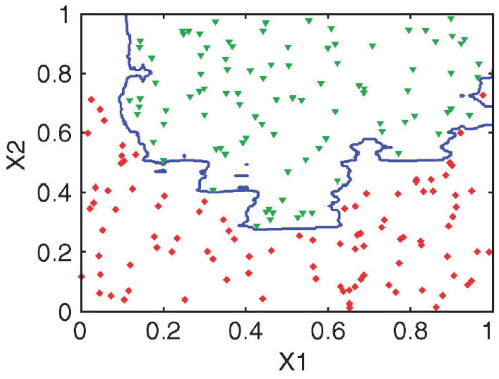
5.4.2 Classification and Regression Trees with AdaBoost

The use of CART in an AdaBoost algorithm can be used to produce more effective weak classifiers. In our examination of the AdaBoost algorithm in section 5.2 we used simple binary classifiers in a two dimentional space, with two possible data labels. As is shown in figure 5.3, even a simple Classification tree can be used to produce a significant improvement in the accuracy of a simple classifier in a two label data space in comparison to a simple binary separation.

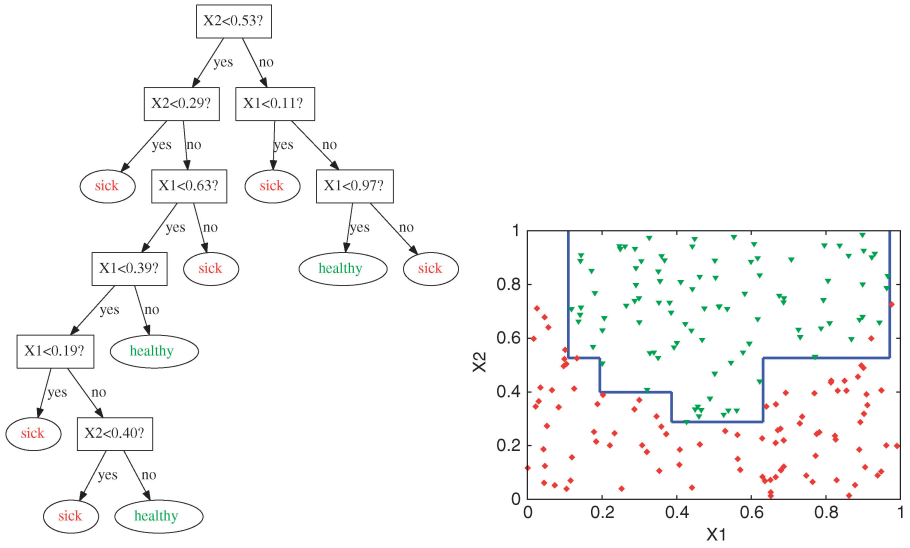
In CART each leaf represents its own simple classifier that can be used by the Ada Boost Algorithm, and in this sense it is possible to think of the example shown in section 5.2 as using a specific kind of tree known as a "stump." (A stump being a decision tree which does not branch, but ends only in leaves.) In both of our AdaBoost example and the CART example in figure 5.3, we are working in a two label space, but this method of labelling is not intrinsic to either model. If we wished to label our data space with a larger number of classifications, CART represents an easy way to construct simple classifiers for a larger number of labels. Figure 5.4 also suggests that it would be possible to have CART classifiers which allow for different subsets of labels within the same data space.



(a) Example DataSpace



(b) CART Overfitting



(c) CART Proper fit

Figure 5.5: Like all Predictive Models CART can be subject to overfitting. The Data space as shown in 5.5a will readily fit into the separations defined in 5.5b and 5.5c, but 5.5b was produced using a tree with over 400 leaves, as opposed to the 9 in 5.5c.

Chapter 6

Experimental Data Set and Test images

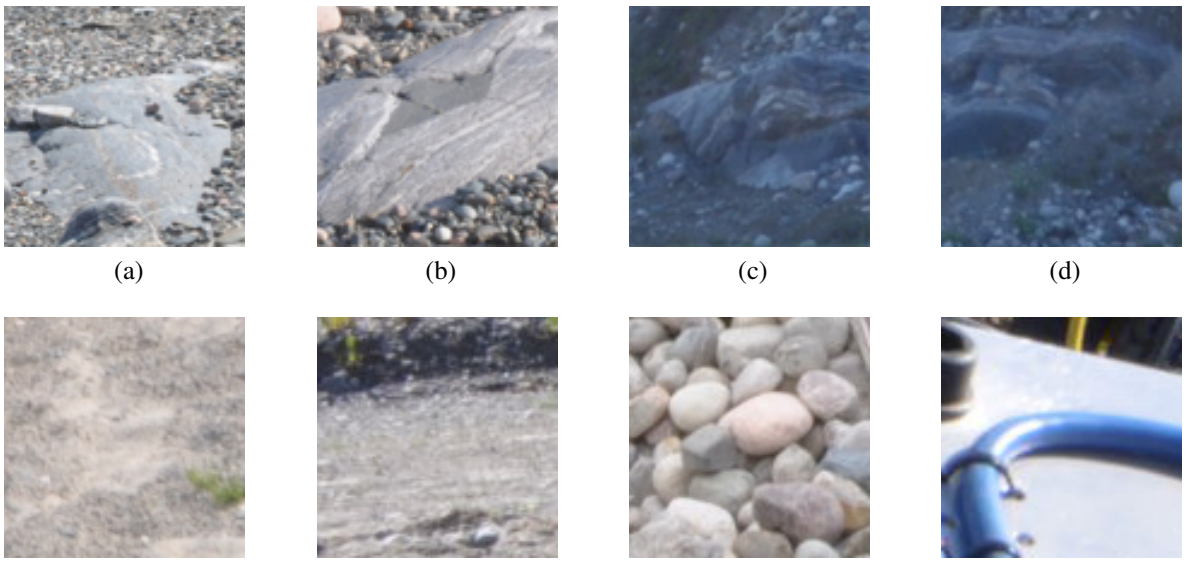
(a) ROC6 Rover (b) GigaPan Camera Mount

Figure 6.1: Images were gathered using a Gigapan Camera Mount attached to the ROC6 Rover.

6.1 Creation of Datasets

In order to properly investigate a machine learning algorithm, it is important to develop a robust data set to be used for training purposes. The images used in the training procedure should, as much as is possible, be representative of the images that will be encountered by the classification committee without being too specific to the task at hand. In each of the two data sets required, this proved to be a particularly difficult problem. In the case of an extraterrestrial environment it is almost impossible to get a sufficiently large dataset from in situ features because of the sparsity of the investigated terrain. While there is a growing dataset of planetary images, the variety of instrumentation, the lack of feature indexing, and the relatively specialized way such images are stored and classified make creating a database from existing archives next to impossible. This limitation required the use of terrestrial analogues for this project.

6.2 Outcrop Dataset



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 6.2: Figures a-d show examples of geological features labeled as interesting, while e-h were considered uninteresting.

The images in the Outcrop Dataset were collected in 2011 at the Ethier Sand & Gravel Pit by members of the Centre for Planetary Science and Exploration (CPSX) for a science simulation mission (SLAM '11) funded by the Canadian Space Agency. All images used were collected remotely using the ROC6 Rover built by the Autonomous Space Robotics Lab at the University of Toronto.

All images used were taken as panoramas from the ROC6 Giga pan imaging mount. These images were separated into two groups. One was for developing the training data set, and the second was used for testing. The training images selected were then divided into three separate groups. From the first group of images, 200, $(n \times n)$ image windows were manually selected from larger images to be used as a training data set. Each image was then resized to be 100×100 pixels large. Half of which contained the desired outcrops and were labeled interesting, the second half were taken from a variety of non outcrop image elements and were labeled uninteresting. These 200 images were then used to train a committee classifier from the complete set of image features described in Chapter 4.

The resultant boosted committee was then tested on the complete images that the training images were taken from. The results of that classification was analyzed and sub windows which were categorized with false positive, and false negative labelling were collected and added to the training database. At the same time those images which were correctly but weakly identified were also added and replaced training examples that were most strongly classified. This step was performed to increase the number of edge cases, to help better define the separation between our two labelings.

We then ran another iteration of the boosting algorithm. The resultant trained classification

committee was then used on the second group of test images, and a selection of weakly correctly identified, positive and negative examples were added to our data set. We also added the most weakly weighted misclassified false positive and false negative results. This resulted in approximately 300 images which were used as a training set on which to run a third iteration of our boosting algorithm.

With the third set of images, we tried a variety of combinations of reduced image features based on the outcome of the previous boosting set. By selecting those image features with the most influence we attempted to minimize the data space we had to traverse to differentiate the data. After the reduced set was decided on, the resultant trained committee was used on a selection of the training images, and any incorrect classifications were added to the training set with appropriate labelling.

The final Outcrop Data Set consisted of 332 100×100 pixel images, 166 of the images were classified as interesting and 166 were classified as uninteresting. An object was considered to be interesting if it was sufficiently large, part of a homogenous structure, and contained a cohesive structure. A large variety of uninteresting features were used as the uninteresting data set.

It is important to note, that in this kind of classification the number of uninteresting examples far exceeded the interesting ones by at least one or two orders of magnitude, and so at each step we discarded approximately 10% of the training sub-images that were most strongly classified as uninteresting for those which were relatively weakly correctly identified as uninteresting, in an attempt to make the differentiation as robust as possible by increasing the number of borderline cases.

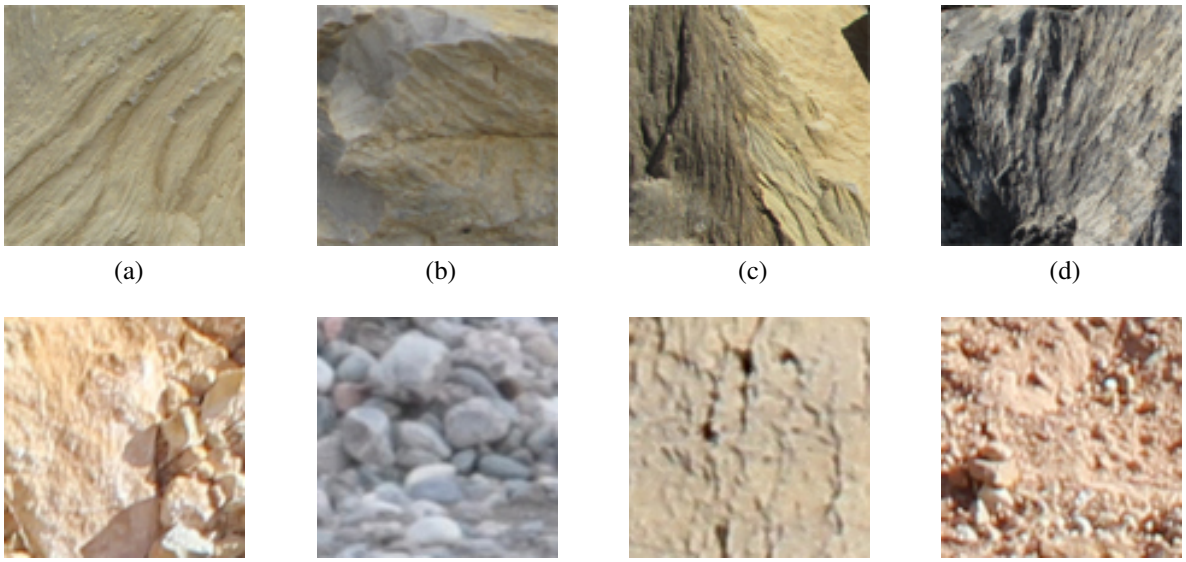
6.3 Shatter Cone Dataset

Examples in the training database were collected from images taken by Dr. Gordon Osinski from the department of Earth Sciences at the University of Western Ontario. The images themselves were taken predominantly from the Prince Albert Impact Crater on Victoria Island NWT. From these original images 80 positive and 80 negative examples were chosen in sub-windows of size $x \times x$ and then resized to be 100×100 pixels in size.

This training set was then used to train a committee classifier from the complete set of image features described in Chapter 4. The results were then tested on the images the features were taken from, and false positive, and false negative results were collected and added to the training database, with weakly valued incorrect classifications replacing those images which were most strongly correctly identified. This process was repeated in a similar manor to the outcrop dataset described above.

A second round of training was done on three images obtained from a Google image search showing complete outcrop structures from the Sudbury Impact Basin as well as some additional examples from the Prince Albert impact crater. The process of adding misclassified images was repeated this time without replacement to produce a larger training set. Lastly a reduced feature vector set was tested against novel images of the Sudbury impact crater based on those image features which performed best and the false classifications were added to the training database.

As with the Outcrop database, the variety and number of negative examples available is far larger and more diverse than positive examples so similarly we replaced approximately 10%



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 6.3: Figures a-d show examples of geological features labeled as interesting, while e-h were considered uninteresting.

of the most distinguishable negative examples with a number of more marginal ones at each iteration. The Shatter Cone Dataset consists of 226 images, 110 labeled interesting and 116 labelled as uninteresting.

Chapter 7

Experimental Results

The composition and attainment of the two datasets drove a decision early on in our research to focus on two slightly different objectives for each of the two classification targets. While the feature set available to be used to begin with was the same, each objective was tested separately with different combinations of those features to optimize the final feature set used in the identification algorithms.

Because the Outcrop dataset was collected from a single locale, on a small number of examples, the risk was that an overly specific training set would limit the generality of the features selected. Based on this fact, the focus of identification was shifted slightly to examine whether a limited dataset could be used to generalize within a region for specific common elements within the landscape. This meant that only part of the landforms captured were used to create the training database, such that other similar structures within the same general site would serve as test examples to attempt to create an effective method of identifying a key feature rather rapidly with limited variety in training examples. The hope was that in an exploration mission a classification algorithm could be developed from relatively few images to allow for autonomous identification as the region was explored.

The Shatter-cone dataset on the other hand while also collected largely from single location had a much larger number of unique examples as well as an expectation of more universal structures existing within examples of shatter-cones around the world. The "fractal like" structure meant that different scales were more flexible to analysis by consensus meaning that a number of smaller regions could be clustered to identify larger structures. These factors make it possible to design our identification algorithms for a more universal applicability.

7.1 Experimental Procedure

The work flow used in both cases can be broken down into 4 distinct and independent steps: Feature Extraction, Training, Classification, and Analysis. While there were slight differences between classification targets they both followed this general pattern. The program was implemented in MATLAB to allow for the use of the VLFeat, and GML AdaBoost Matlab Toolbox libraries.

7.1.1 Feature Extraction

Feature	Implimentation Used
SIFT	vl_dsift() from the VLFeat Library.
Haar	Custom regions using MATLABs cumsum() function.
HOG	vl_hog() from the VLFeat Library
Edge Orientation Gradients(EOG)	Custom algorithm.
Hough Transforms	MATLABs hough(), houghpeaks(), and houghlines()
Intensity Histograms	MATLABs imhist() function.
Edge Density	Custom algorithm and feature.

Table 7.1: A number of available feature implementations were used to simplify coding. The extracted features were then processed in order to form useful descriptors which could be inserted into the feature vector

SIFT SIFT features are extracted using a *dense SIFT* algorithm with a spacial bin size of 3 pixels every 2 pixels. We then divide the image into 4 layers vertically, and 4 layers horizontally, and combine those 16 regions into 27 different orientation combinations. These combinations are then clustered into ($k = 100$) clusters, yielding a feature vector F_{sift} with a dimensionality of 2700.

Haar-like features Haar features are collected by first creating an integral image of each image. Each image is then divided up using patches of 50×12 pixel sub-windows in both a horizontal and vertical direction in overlapping steps every 5 pixels. Each window computes the sum of 6 different haar features (see figure 7.1.) For a 100×100 pixel image window this generates a feature vector F_{haar} of size $198 \times 2 \times 6 = 2376$.

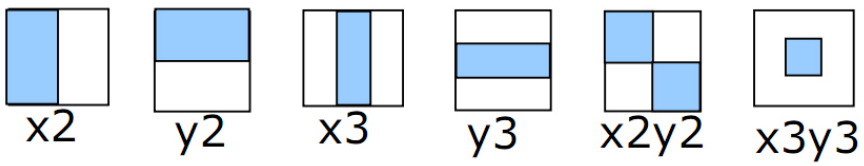


Figure 7.1: The six Haar-like features.

HOG For HOG features, five different bin sizes(B) were chosen where $B_i = 2, 5, 10, 20, 25$ Each histogram had 9 histogram orientations equally spaced between 0° and 180°. For a 100×100 pixel image this produces 5 $x \times y \times 31$ matrices(P) where x and y are the number of bins required to traverse the image in non overlapping segments. In the case of our 100×100 pixel images, the size of x and y are determined by $x_i = y_i = 100 \div B_i$. This results in 5 matrices, P_i where $x_i = y_i = \{50, 20, 10, 5, 2\}$respectively. The results are concatenated to produce a feature

space P_{all} where P_{all} is a 3041×31 matrix. We then cluster the matrix into 100 bins giving us a feature vector F_{hog} of size 100.

Edge Orientation Gradients To calculate the Edge Orientation Gradients we use a single band image i of the image. The image is divided into 4×4 non overlapping quadrants ($\{i_{1,1}, i_{1,2}, \ldots, i_{4,4}\}$), and in each region we calculate a 1×5 histogram orientation vector h_i consisting of 5 orientation parameters (horizontal, vertical, 2 diagonals, and 1 non-directional.) This gives us a feature vector F_{EOG} of size $4 \times 4 \times 5 = 80$

Hough Transforms Two different image features were extracted from the Hough transform, Hough peaks and Hough lines. starting with the original image i, we perform a canny edge detection to get ic. A Hough transform is performed on the edges, resulting in a hough Matrix H of size $\theta \times \rho$ where $-90 \leq \theta \leq 90$ and $-(\rho_{max}) \leq \rho \leq \rho_{max}$ where ρ_{max} is determined by $\rho_{max} = \sqrt{ic^2_{length} + ic^2_{width}}$. Using H, up to 500 of the largest peaks are found with a minimum amplitude of $0.7 \times H_{max}$. The average energy of the euclidian distance between the points and the distribution centre is calculated and as well as a histogram of the peak distribution over θ, this produces feature vector $F_{HoughPeaks}$ of size 313. Lines longer than 7 pixels are then found, and a 4 dimensional feature vector $F_{HoughLines}$ is calculated from: Maximum line length, sum of all line lengths, the number of lines greater than 7 pixels, and the average length of the lines longer than 7 pixels.

Intensity Histograms Two different intensity histogram features are used, the first feature, $F_{hist_{intent}}$ converts the image i into and calculates a 32bin intensity histogram, the second feature, $F_{hist_{rgb}}$ takes the r, g, and, b channels of i and calculates the histograms of each layer. individually. Further, i is divided into 4 non overlapping quadrants and a histogram of each quadrant is calculated as well as combinations of these quadrants. This leaves us with $F_{hist_{rgb}}$ of size 1056, and $F_{hist_{intent}}$ of size 352.

Edge Density Edge Density is calculated by using a Canny edge detector on the image to locate edge pixels. The Canny image (ic) is divided into nine rows and columns of and each edge pixel is then summed giving $ic_{row_1}, ic_{row_2}, \ldots, ic_{row_9}, ic_{col_1}, ic_{col_2}, \ldots, ic_{col_9}$. 3×3 non-overlapping squares of ic are also calculated (imagine a Sudoku board.) each area is summed and a histogram count is then made of rows, columns and patches to produce an edge density feature $F_{EdgeDensity}$ of size 270.

7.1.2 Training

In order to train the decision committee after the feature vector created from the above image features, we then add a label to signify that the image patch is considered either interesting or uninteresting. This label is represented by a single digit (1 or 0) at the end of the feature vector. Once the interesting and uninteresting feature vectors have been constructed they are then passed to the AdaBoost training algorithm.

In our work we experimented with CART, using hypotheses consisting of a tree with node size of 1, 2, 4, 8, and 16 nodes. In each case the data was divided into cross validation folds,

(a number of different sizes of cross validation folds were tried, with 3 folds seeming to best represent a balance between bias and variance errors.) Once separated, two folds were used for training while the other was used for testing. This process is repeated for each fold. The training utilized three separate boosting algorithms in parallel, Real AdaBoost, Gentle AdaBoost, and Modest AdaBoost. Each algorithm has its own advantages, and so we began using all three as a way to compare the effectiveness of each, and found that as different combinations of features were used, the performance of any given algorithm changed with respect to the others. Ultimately all three algorithms were used to train separate committees to help form a consensus as we will see in section 7.1.3. The number of iterations of the boosting algorithms tried ranged from as few as 200 to as many as 2000 with 400 seeming to be sufficient to ensure minimum test error without overfitting.

Feature vector	Combined Vector		Test Accuracy		F_{size}
	Outcrops	Shatter-cones	Outcrop	Shatter-cones	
F_{sift}	No	Yes	68%	72%	2700
F_{haar}	Yes	Yes	72%	60%	2376
F_{HOG}	No	Yes	61%	58%	100
F_{EOG}	No	Yes	75%	77%	80
$F_{HoughPeaks}$	Yes	Yes	75%	68%	313
$F_{HoughLines}$	Yes	Yes	62%	73%	4
$F_{hist_{inten}}$	No	Yes	62%	53%	352
$F_{hist_{rgb}}$	Yes	No	85%	83%	1056
$F_{EdgeDensity}$	No	Yes	72%	71%	270

Table 7.2: Overview of feature effectiveness

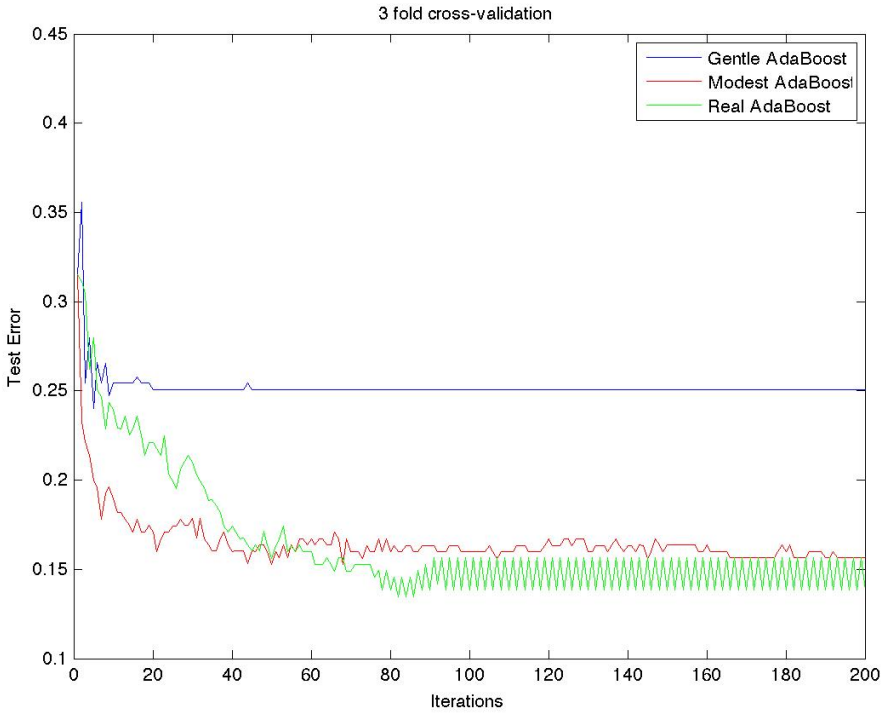
7.1.3 Classification

The process of classifying an image begins by using an $n \times n$ detection window with a stride/offset (o) of between 25 and 33%. depending on the window size. Various window sizes of $n = 800, 600, 400, 300, 200, 100, 64, 24$ were tried. Windows are then interpolated to 100×100 squares. From these squares a feature vector is extracted for each and then given a value based on the discriminant function developed from the Boosting algorithms. This meant that each detection window sub-image is given is then given three weightings. Since the stride was as little as 25% each image patch could be given up to 48 weighted votes.

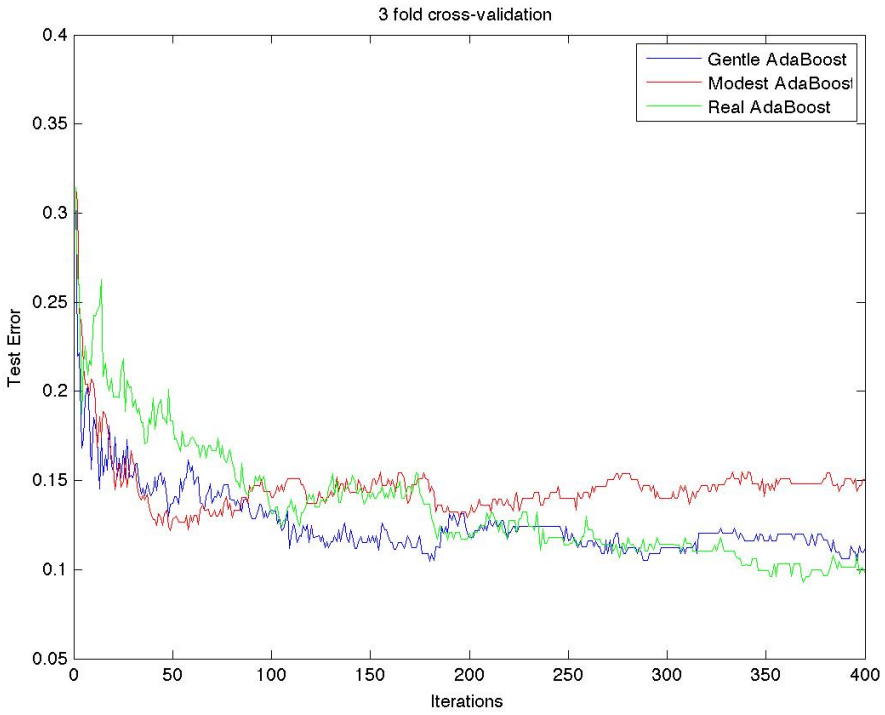
7.1.4 Analysis

Analyzing the outcome of image features is done through a statistical consensus of the values obtained from classification step, and consists of three individual components which are combined to determine the likelihood of a desired feature existing in any given image.

Weighted Vote The results of the discriminant functions generated by each of the boosting algorithms give both a label (+ or -) and a confidence weighting (represented as some real

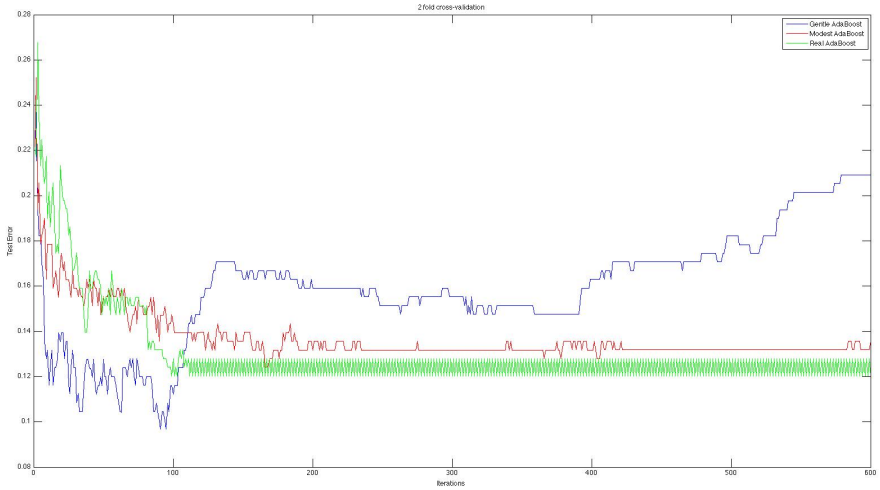


(a) Training with a decision tree with 16 nodes

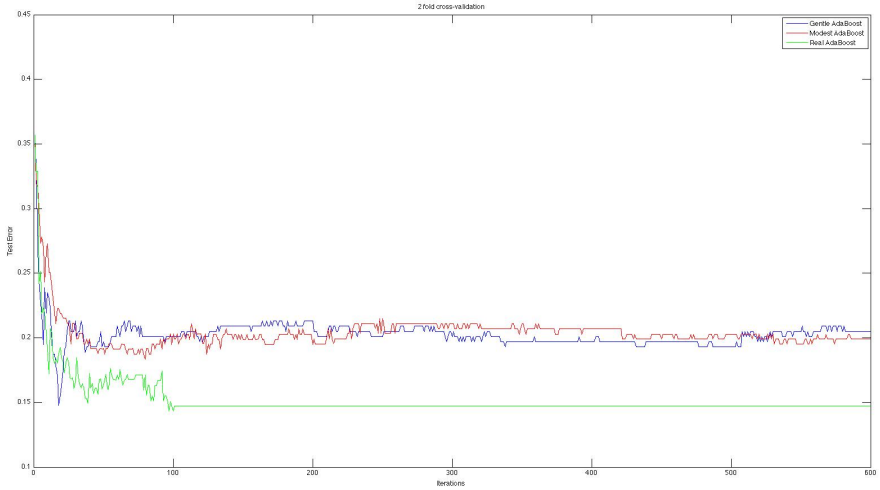


(b) Same Features as (a) but using 4 nodes

Figure 7.2: We can see in figure 7.2a and 7.2b that the number of nodes had a dramatic effect on the outcome of the training with Gentle AdaBoost, while Modest AdaBoost had very similar results.

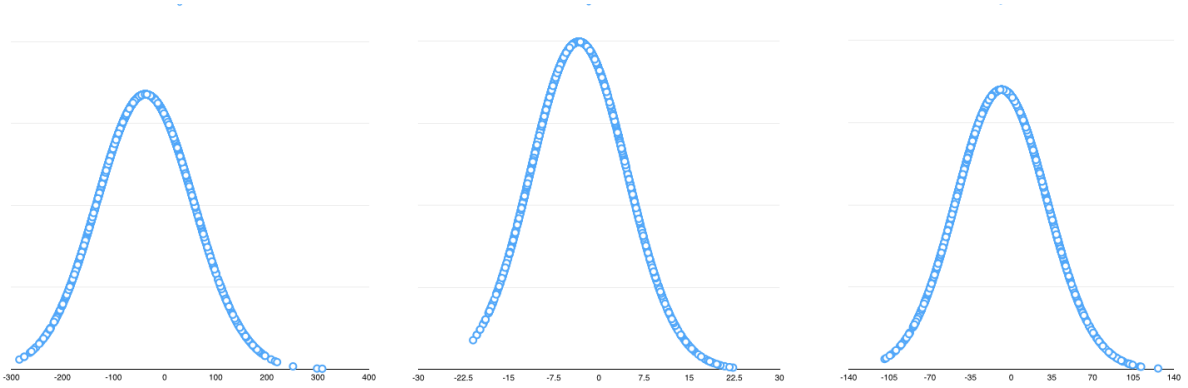


(a) 8 Node Tree



(b) also an 8 node tree with the same features as (c)

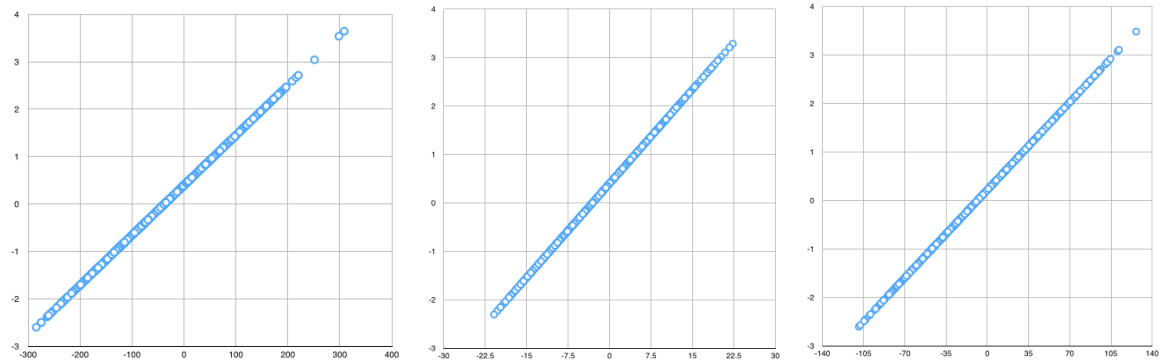
Figure 7.3: In 7.3a and 7.3b we find that the same features produced different results during different runs because of the variation in cross validation folds.



(a) Gentle AdaBoost Distribution (b) Modest AdaBoost Distribution (c) Real AdaBoost Distribution

Figure 7.4: Each boosting algorithm produces different confidence amplitudes for the same image. In order to produce a weighted vote these confidence ranges need to be normalized. The distribution shapes are slightly different and so the standard scores of each are used.

number.) In cases where a single boosting algorithm is used, it would be possible to determine a weighted vote from from this confidence value alone. Unfortunately, as can be seen in Figure 7.4, the confidence values assigned by each boosting algorithm vary widely. Since the distributions are not identical and any outliers may skew amplitude normalization, we use the standard score to produce a weighted vote as seen in Figure 7.5. In calculating the votes, we only calculate those image windows with a positive label value, and we do not include any negative standard scores.

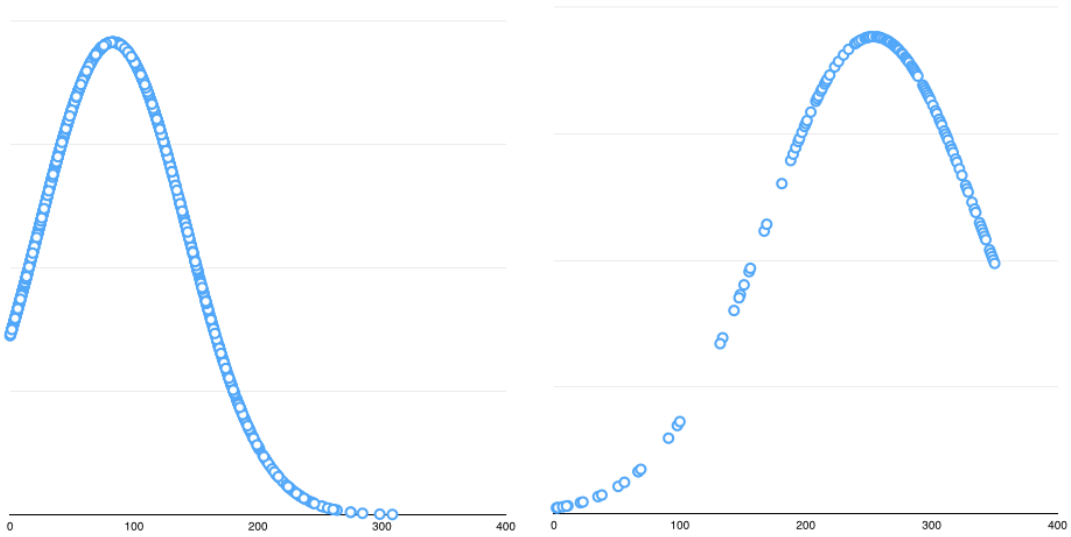


(a) Gentle Standard Score (b) Modest Standard Score (c) Real Standard Score

Figure 7.5: The standard score of each distribution is used to generate a weighted vote.

Global Image Confidence In addition to a weighted vote for each region we include a confidence weighting which is calculated by using the distribution of amplitude values boosted discriminant function. This is done in order to wight the overall confidence of our data model on a given image. The amplitude values in a poorly differentiated image will skew towards lower amplitudes as in Figure 7.6a while strongly differentiated images will skew towards

higher amplitude values as seen in Figure 7.6b. By using the number of standard deviations zero is from the mean in each image we can produce a global confidence value for each image.



(a) Weakly differentiated features in an image (b) Strongly differentiated features in an image

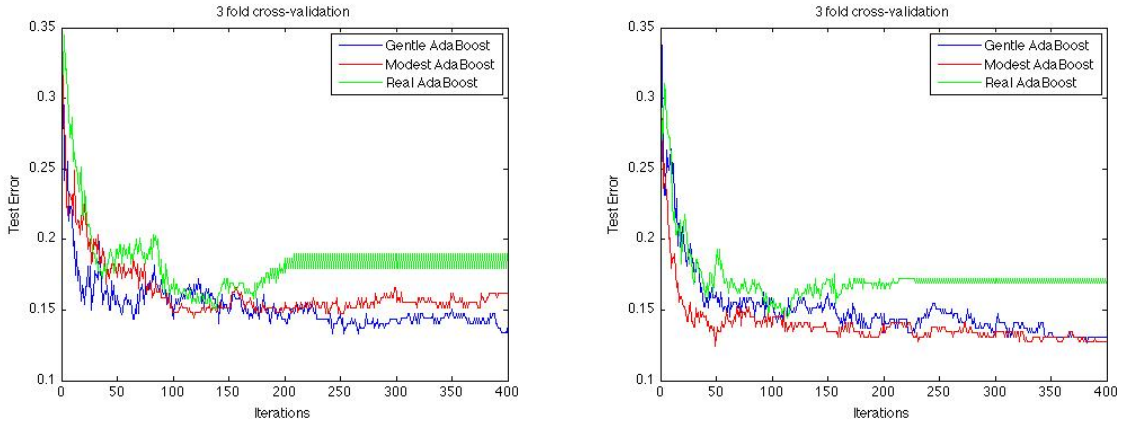
Figure 7.6: A data model that produces poorly differentiated data will skew toward zero, while strongly differentiated data will skew towards a higher amplitude values

Confidence Distribution Once each image region has been given a value based on the weighted votes, we can generate a confidence likelihood of something of interest existing by normalizing the votes with the global image weight we can produce a confidence distribution across each image. The windows in Figure 7.8a shows those regions that have been given a positive rating by the boosted committee's discriminant function. Figure 7.8b shows the confidence distribution of positively identified features with warmer colours representing patches with the highest weighed values.

7.2 Outcome of Outcrop Classification

In the process of building the training database it was noted that a subset of features gave results equivalent to, or perhaps slightly better than the full image feature Vector (See Figure 7.7.) Because of this our final testing used only a subset of the image features, specifically: Haar-like features, RGB intensity histogram, and Hough transform (both Hough Peaks and Hough Lines.) This reduced data set significantly improved the speed at which the image vectors could be processed for each sub-window, as well as an improvement in the AdaBoost algorithm training time which had a much smaller feature vector to iterate through.

To test the effectiveness of our algorithms, we devised 5 separate test scenarios. The first test was to use the complete algorithm on one of the images used in the creation of the database. This test was to test the success of the probability algorithm on a scene where we were confident



(a) Training with all features

(b) Training with only Hough Transform Haar and RGB histograms

Figure 7.7: In 7.7a all features produced a test error of between 12-20% after 400 iterations. A subset of features achieved a Test Error rate of 12-17% after the same number of iterations.

of correct sub-window identification (see Figures 7.8a and 7.8b.) It also offered an opportunity to identify more challenging negative examples within the test set imagery

The second test scenario used rock features that were included in our training data set, but on images that were taken from a different location, on a different date, with slightly different lighting conditions. This test was designed to test the effectiveness of the classification committees ability to classify known features (see Figures 7.8c and 7.8d.)

The third test was a completely novel image taken with a separate camera that did not correspond to the geometry of the original images in order to test how instrument specific our model was. In order to test how applicable our sample of uninteresting training examples were, it also introduced some unique ”uninteresting features,” by way of using images which included human beings, tents, trucks, a toy sword, and the ROC-6 rover, which had not been included in the training set at all (see figure 7.8e.)

The fourth test was to test our statistical method against an image with an unusually large number of positive features or an unusually low number of positive features, such that it should present a difficulty to our statistical likelihood algorithm by having a skewed distribution of sub-windows which gave a positive or negative result. We did this to test if a significantly unusual probability spectrum would introduce false identifications, or miss significant regions (see figure 7.10a.)

The fifth test was to apply our decision committee to completely separate outcrops outside of the test site to see if our algorithms translated at all, and if so how well and how consistently they did. Ultimately this test was outside of our initial goals, but was undertaken to try and probe the limits of our method. We conducted this test with little expectation of success, however we feel the results were interesting enough to merit a discussion.

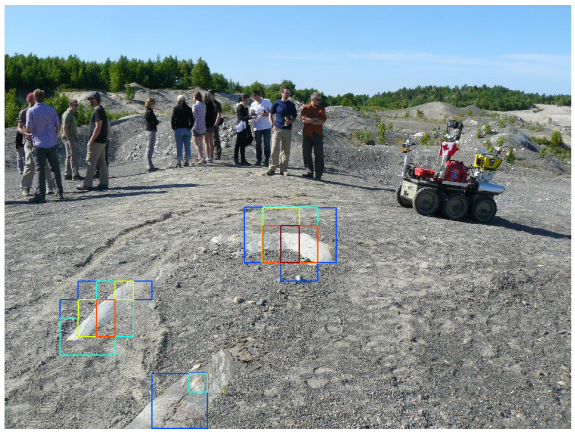
Our ability to detect known regions in the first series of tests was quite successful, which was expected as these features were part of our training data set. For all of the images from which we pulled image features, we were able to reasonably detect the outcrops. Because we did not focus on the exact dimensions of an outcrop we considered it a success if probability

(a) Test 1 Result

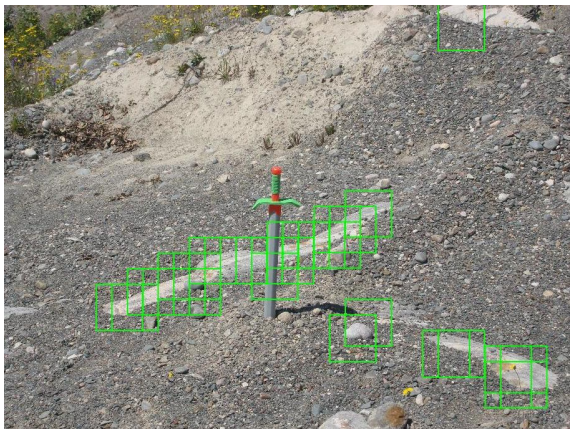
(b) Probability map of (a)

(c) Test 2 Result

(d) Test 2 Result



(e) Test 3 Success



(f) Test 3 Result

Figure 7.8: Successful Identifications of Outcrops

peaks existed within the confines of the outcrop in the image.

It was noted during this first test that the size of the sub-window used had a significant impact on the identification rates. Despite this fact, the probability analysis was able to ignore isolated positive identifications (such as can be seen at the left side of Figure 7.8a) It may also mean that distant features which only show at the smallest scale might be missed. The implications of this will be discussed further in Chapter 8.

The second set of tests also performed to our expectations with all images having correct labelling as was seen with the first test. All four images used resulted in the correct identification of the outcrop with the highest probability.

The third test produced mixed results; Most of the novel images produced successful or largely successful results. However one notable exception was a photograph from a ledge which contained no immediate foreground (figure 7.9a.) As a result of either a failure in the training dataset or a failure in our algorithm, an inordinately large number of false positives were generated which overwhelmed our attempt to smooth it with our probability function.

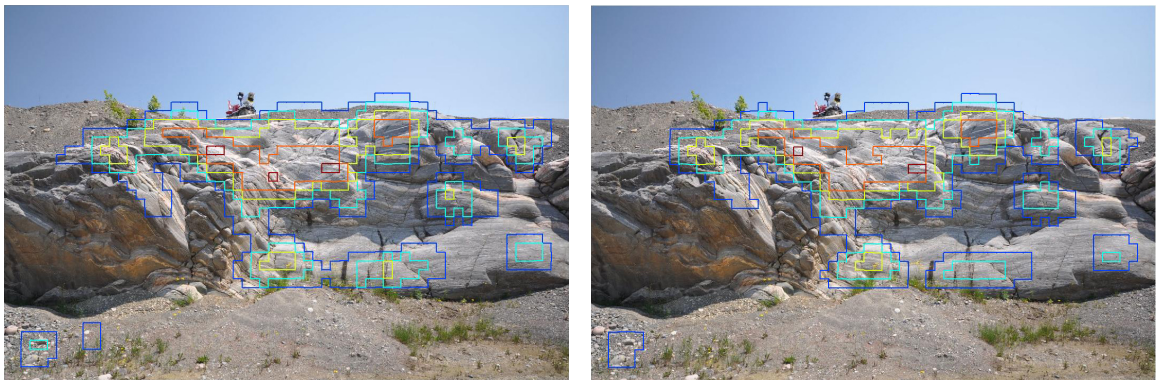
While the desired outcrops on the middle right hand side of the picture were identified, two key failures were noted. First at the bottom of the image, a rock pile is given the strongest probability of being an outcrop, while in the far background at the upper and middle right, sand piles were falsely identified as being outcrops. This we believe is due in part because of the resolution of the features detected. The sand piles at the rear of the image are so far away that they appear as homogenous structures without the foreknowledge of their extreme distance, while the rocks in the pile at the bottom of the image may appear to be homogenous to larger window sizes which skewed our image voting mechanism. This issue may present some difficulty in the translation of this work to extraterrestrial environments, and we will discuss this possibility in Chapter 8.

(a) Test 3 Novel Image (b) Test 3 Result

Figure 7.9: When applied to a distant perspective, the algorithm or possibly the training database failed to produce a useful result.

The fourth test used images which contained either no relevant features, or were dominated by relevant features. Originally we designed this test to test whether the statistical model we designed would significantly change the outcome if the distribution of expected objects was

skewed one way or the other. While we were fairly confident that large amounts of uninteresting sub-windows would not be problematic because we use a minimum value threshold in our voting procedure, we did worry that a large number of positive sub windows would skew the distribution such that it would significantly alter our voting procedure. As we can see from the comparison of probability maps in figure 7.10, the use of an simple yes/no heuristic produced a very similar result to our statistical distribution method, while our method seems to produce much better results in areas of localized features.



(a) Test 4 Using Proability (b) Test 4 Using Committee Values

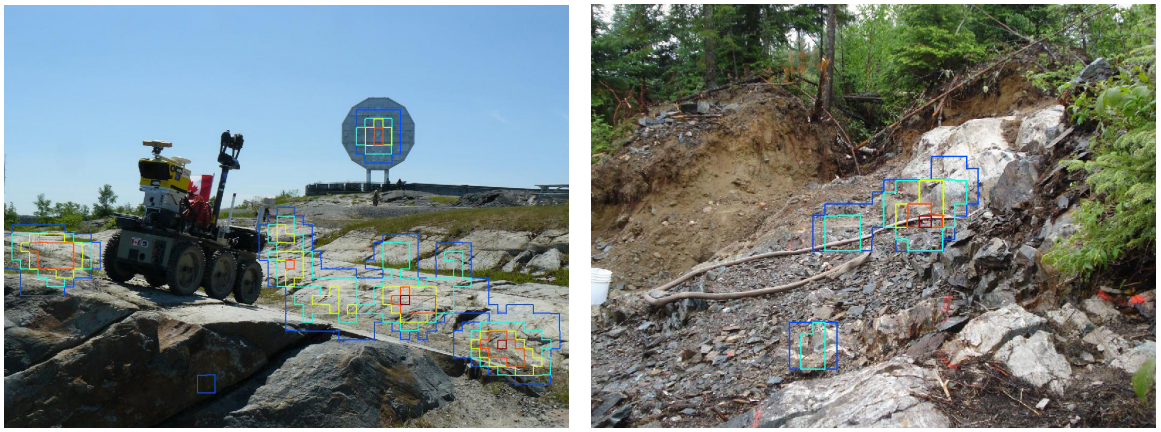
Figure 7.10: Above we see a comparison of our Probability Method of voting compared to using the boosted committee to generate a probability map. 7.10a shows the probability map of our method, while 7.10b was generated by giving a vote to any sub-window which gave a positive identification.

Our fifth and final test was to apply our results to completely novel outcrops, and situations to see if the resultant classifications were comparable. The results we had were somewhat surprising in that our algorithm performed better than we originally expected, but also seemed to translate relatively well to surrounding rock structures fairly consistently. It should be cautioned that we did not exhaustively test the reasons for this difference, those outcrops we could find which were located in the same impact structure seemed to give better results than those of completely separate structures (see figure 7.11.) While there a fairly significant degradation in accuracy from leaving the immediate area, it does suggest that the classifiers may be robust enough to enable an iterative generation of a training set during the course of a mission that traverses relatively large distances as is the case with the MER rovers.

7.3 Outcome of Shatter-cone Classification

7.3.1 Training and Control Error

Our approach to the classification of shatter-cones parallels the methodology used for the classification of outcrops with some adaption for the differences in structure and usefulness. To begin with the building of the shatter-cone training database showed the usefulness of the inclusion of all of the features we explored previously with the exception of the RGB histogram.

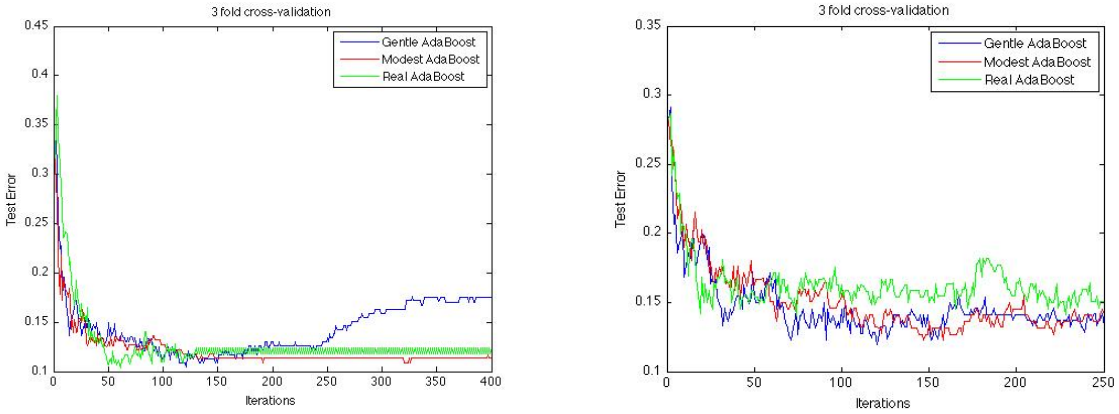


(a) Test 5 using a local rock structure

(b) Test 5 using a random outcrop

Figure 7.11: Above we see a comparison of the trained committee applied to completely novel outcrops. 7.10a shows an outcrop at a different location within the Sudbury Impact structure, while 7.10b shows the result of our classifiers on an geographically unrelated outcrop. While we did not extensively test this phenomena, there seems to be a better transition of our features to structurally similar outcrops in comparison to outcrops which are simply visually similar.

While the inclusion of the RGB histogram did produce lower error rates for shatter-cones within a given training data set, the difference was not significantly large (see figures 7.12a and 7.12b.) The inclusion of the histogram gave on average a lower relative weighting of shatter-cones contained in different impact craters located in different rock types, particularly when a variety of materials were present in the image. We will discuss this decision in greater detail in Chapter 8.



(a) Test Error All Features

(b) Test Error All but RGB

Figure 7.12: Training results using a full set of features with and without the RGB histogram

To test the classification of our Shatter-cone images, we needed to tackle a number of requirements to prove the applicability of a boosted training dataset across as broad a set of conditions as possible. The first requirement is that the trained committee, which was trained

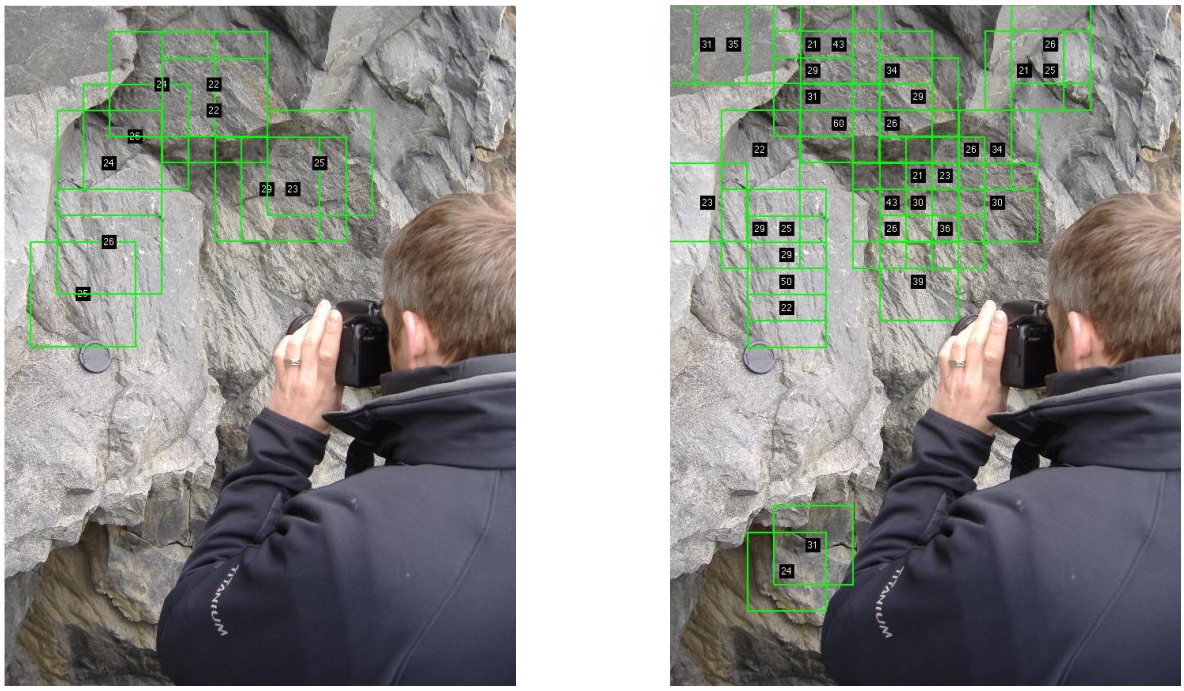
from both complete shatter-cones and small subsets of larger shatter-cones, should scale to identify the entire cone clusters within the same image when at different scale. The second requirement is that a the committee should be able to identify shatter-cone formations that had affine translations to the image plane as well as shatter-cones interspersed within different formations. The third requirement is that the committee should be able to identify shatter-cones found in other rock types, with differing compositions. In order to test these requirements we devised four scenarios under which we would test our algorithm.

The first test consisted of identifying shatter-cones located in the images used in the training data base but utilizing the entire image rather than only a subset of regions. This rst test was to designed to show the both the scalability of the feature set and to rene the training data base. The second test was conducted using images of shatter-cones gathered from the same location but not used in the training data set and was designed to ensure that the features selected would be effective in a variety of imaging conditions. Including shadow, scaling and focus biases. In the third test we used shatter-cones in a variety of locations, conditions and rock types found in images gathered from impact craters from around the world. Our final test was to test the trained committee on landscape scale images to determine how effective our algorithm would be in differentiating the appearance of shatter-cones in novel situations.

7.3.2 Test 1 Familiar Images Different Scales

The only images used in this test were those images which contained the best examples of shatter-cones, and thus were chosen to form part of our training database. This type of test is generally not helpful in identifying the accuracy of our training committees because our committee was based on the same images we are testing it on and thus subject to over fitting. It was determined that this test would be useful for two reasons, first it allowed for a better refinement of the negative examples, allowing removal those negative examples which were most obvious,substituting examples which were closer to the borderline of the two groups, and secondly it allowed us to incorporate as many different positive examples as we could find again emphasizing those closest to the edges of the dividing line so as to best define our models. After performing this first test, false positives and false negatives were swapped into the training data set while strongly weighted true negatives were swapped out in an attempt to better isolate the structural components of shatter-cones in the training database, while at the same time reducing the weighting given to incidental factors.

The results showed a predictably high accuracy in identifying the most prominent shatter-cones in these images. It was found that varying the size of the detection window had a significant influence on the detection of shatter-cones with a strong bias towards those which completely filled the detection window. This meant that often shatter-cones clusters which either changed direction or were partially occluded would not be detected as can be seen in figure 7.13a and 7.13b. Overall detection rates were between 70%-90% with 100x100 pixel detection windows and 50%-80% with 200x200 pixel detection windows. A corresponding increase in false positives was also noted with smaller detection window sizes.



(a) Confidence values using 200px windows

(b) Confidence values using 100px windows

Figure 7.13: The same image run with initial training data base at different section window sizes.

7.3.3 Test 2: Familiar Geology, Novel Images

Test 2 was designed to test how well our trained features performed in more challenging photographic circumstances while attempting to not introduce changes within the actual geological composition of the rocks. Image were selected from the original group collected in Prince Albert Impact Structure which contained more challenging conditions such as those seen in figure 7.14. The images selected contained a variety of lighting conditions and surfaces on an oblique angle to the camera plane.

In addition to the inclusion of more difficult features, a variety of imaging sizes were chosen, with the windowing sizes ranging from 10% of total image size to 3%. (i.e. 200px, and 100px windows were used for figure 7.14 which is a 2400×2000 pixel image.)

The results of our second test showed three informative results. The first is that the boosted decision making committee used was able to identify Shatter-cones in a while variety of orientations and angles relative to the camera plane. The success rates of labelling those features which were not parallel to the image plane were similar to the first test. The caveat here is that this pertains only to well lit features which leads us to our second informative result.

Features in shadowed regions of the image faired far worse than those that were well lit. Although some image sub windows did identify the shadowed feature, (see bottom left of centre in figures 7.15a and 7.15c) the weighting given was significantly lower. While this may have negative implications in applications that require high levels of accuracy, the detrimental effects might easily be mitigated through a number of strategies for our purposes that we will

Figure 7.14: Typical test image used in our second Test round

discuss in more chapter 8.

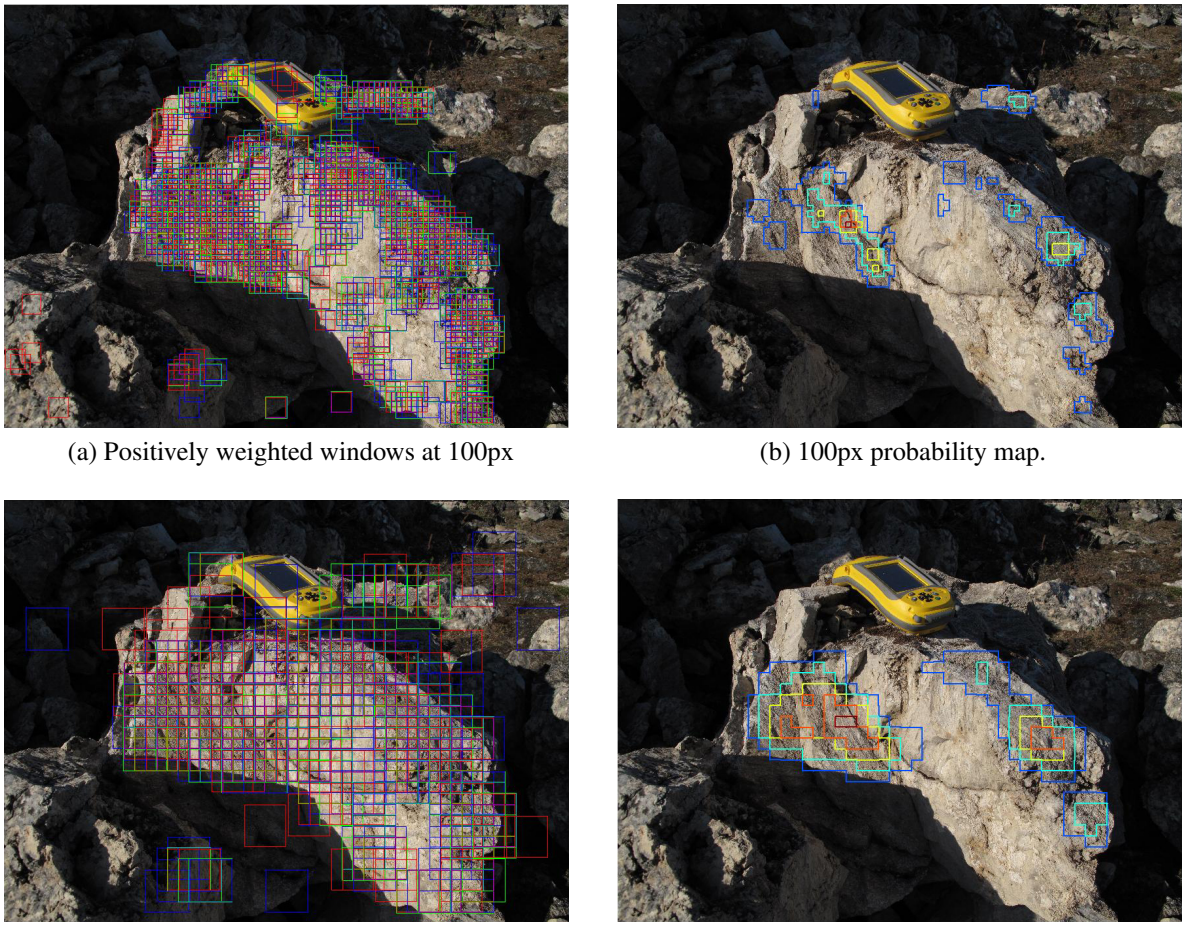
The third result which proved informative was the effect of sub-window size on feature identification. As can be seen in figure 7.15b, the smaller window size allows for the labelling of higher probability of smaller features (see the blue square to the right of the GPS unit) compared to the larger window size (figure 7.15b.) the highest probabilities of both window sizes was centred around the same features.

This result suggests that the smaller window size would provide a more accurate identification of shatter-cones, or at least reduce the number of false negative results of smaller cone structures. Although this is problematic in its own right because the window size has an inverse square relationship with the algorithm running time, which as we will discuss in chapter 8 is one of the most challenging issues of our algorithm. While this is an undesirable result, a comparison of figure 7.15c and figure 7.15a shows that an overwhelming majority of regions with a positive weighting in figure 7.15a are also covered by the positive regions in figure 7.15c. This suggests that it may be beneficial to utilize an iterative pyramid approach with increasingly smaller sub-windows to maximize feature resolution while minimizing processing power.

This result was intriguing enough that on Test 3 we chose to expand the range of window sizes we used to see if the observed trend held true for larger windows.

7.3.4 Image Test Set for use inTests 3 & 4

The images for our third and fourth test were selected from a variety of locations from around the world. In total 30 images were selected from 8 di?erent impact sites. (See table 7.3.) 20 images were selected for use in Test 3, and ten wider angle landscape style shots were chosen for Test 4. For the purposes of this test, images which contained prominent examples



(a) Positively weighted windows at 100px

(b) 100px probability map.

(c) Positively weighted windows at 200px

(d) 200px probability map.

Figure 7.15: A typical test image used in test 2 of the shatter-cone images. This image was selected for shading variations and oblique surfaces relative to the camera plane.

Country	Impact Site	Co-ordinates		Images Used
Canada	Sudbury Impact Basin	46.6000° N,	81.1833° W	6
	Prince Albert Impact Crater	72.4667° N,	113.9333° W	2
France	Rochechouart Impact Structure	45.8242° N,	0.7817° E	4
Germany	Ries Impact Structure	48.8833° N,	0.5667° E	3
	Steinheim Impact Structure	48.6867° N,	10.0650° E	3
S. Africa	Vredefort Dome	27.0000° S,	27.5000° E	6
USA	Crooked Creek Impact Structure	37.8355° N,	91.3950° W	3
	Santa Fe Impact Structure	35.7500° N,	105.9333° W	3

Table 7.3: This table lists the composition of novel images used in testing from a variety of sources. All of the images of Prince Albert Impact Crater were courtesy of Dr. Gordon Osinski, while the images of Vriedfort Dome are courtesy of Professor Pierre Thomas of Laboratoire de Geologie de Lyon. Images from the Sudbury impact basin come from the Western Earth Sciences 200a website [15]. The rest of the images were found on the Shatter-cone Page of Dr. Kord Ernstson and Fernando Claudin [26].

of shatter-cones were chosen in order to identify how universally applicable our method was. Images were chosen to provide a variety of rock types and colours. An attempt was also made to find a variety of sizes and formation completeness while eschewing both artificially lit and physically isolated specimens . Wherever possible the images chosen contained only mineral material although this proved next to impossible for the landscape scale images chosen for Test 4.

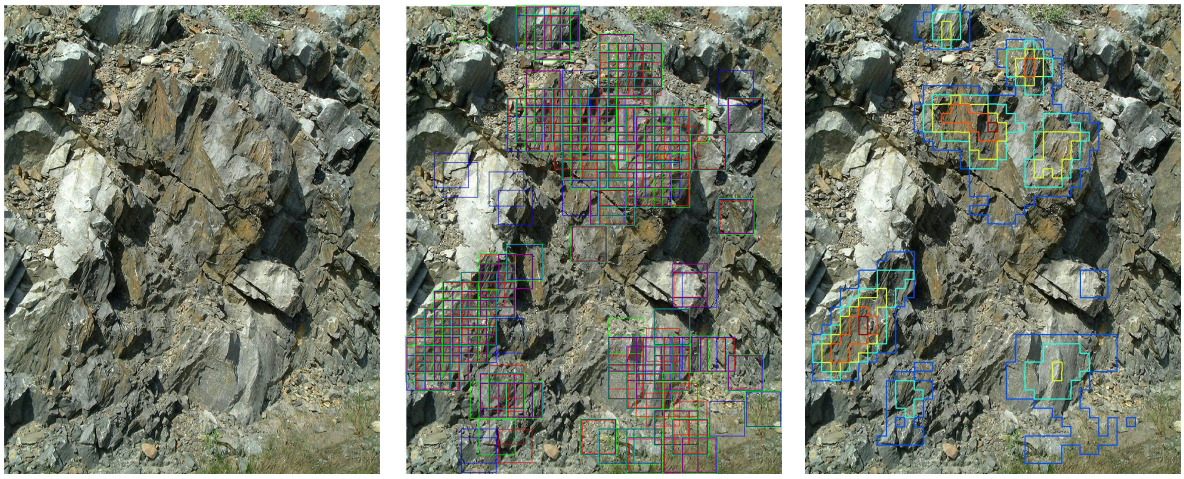
7.3.5 Test 3 Novel Geography, Limited Scope

The results from Test 2 meant that Test 3 was broken down into two separate but connected avenues of exploration. The first was a determination of the efficacy of our methodology in identifying shatter-cones that differ in appearance and composition from our training group. The procedure for this test was identical to our previous two, and a representative result can be seen in Figure 7.16. The results we obtained showed identification rates similar to those found with the novel images from the Prince Albert Impact Crater which were used in the previous test. This suggests that the features we chose translate well to different rock types. The introduction of cracks debris lead to some false identification, but much of this was eliminated by weighting introduced by the probability map.

The second avenue of exploration was an examination of the effects of window size on the localization of shatter-cones. To test this the range of window sizes that were tried ranged from 24-400 pixel squares. The results from this test showed that as expected the smaller window sizes were better at detecting smaller shatter-cones while still being able to partially identify parts of larger structures due to the "fractal" nature of shatter-cones. At larger sizes, the windows were able to identify large features, while missing out on finer features. The number of false positives also increased. As can be seen in figures 7.18 . While this suggests that there is some merit in a multiple pass pyramiding approach of windowing, it raises some concerns as well which we will discuss in Chapter 8.

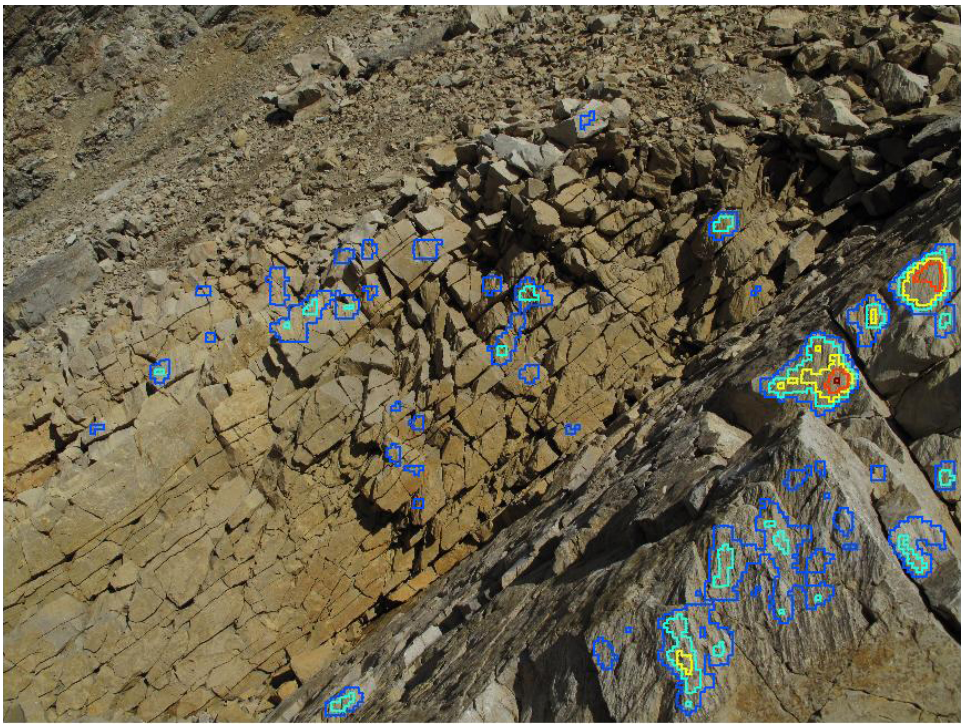
7.3.6 Test 4: Landscape images

The construction of test imagery for the last test proved unexpectedly difficult as in general images which are labeled as shatter-cones consist largely of framed images of the shatter-cones themselves, and those which might contain shatter-cones as part of a larger image tend to be rarely labeled as having shatter-cones. The images in figure 7.19 are representative of the types of images used in this test. These images contain confounding factors not expected in an extraterrestrial setting, which proved to cause some difficulty for our algorithm. In figure 7.19a for example the strands and cone like shape of the grass tufts confused the algorithm. In figures 7.19b and 7.19c the presence of people gave the program pause. Despite this fact the algorithm did put the highest probability on shatter-cones in the image.

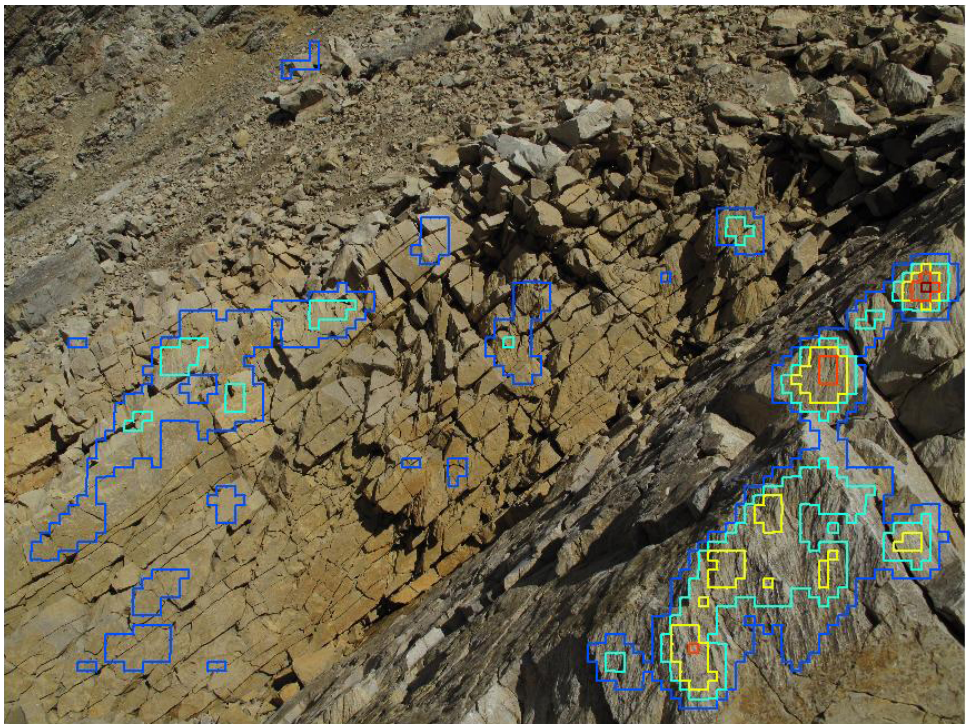


(a) Shatter-cones in Sudbury Breccia (b) Identification of Shatter-cones (c) Labeled Shatter-cones

Figure 7.16: From Left to Right: The original image. Windows labeled by our algorithm as shatter-cones. A weighted distribution of probability that the encircled image contains a shatter-cone.

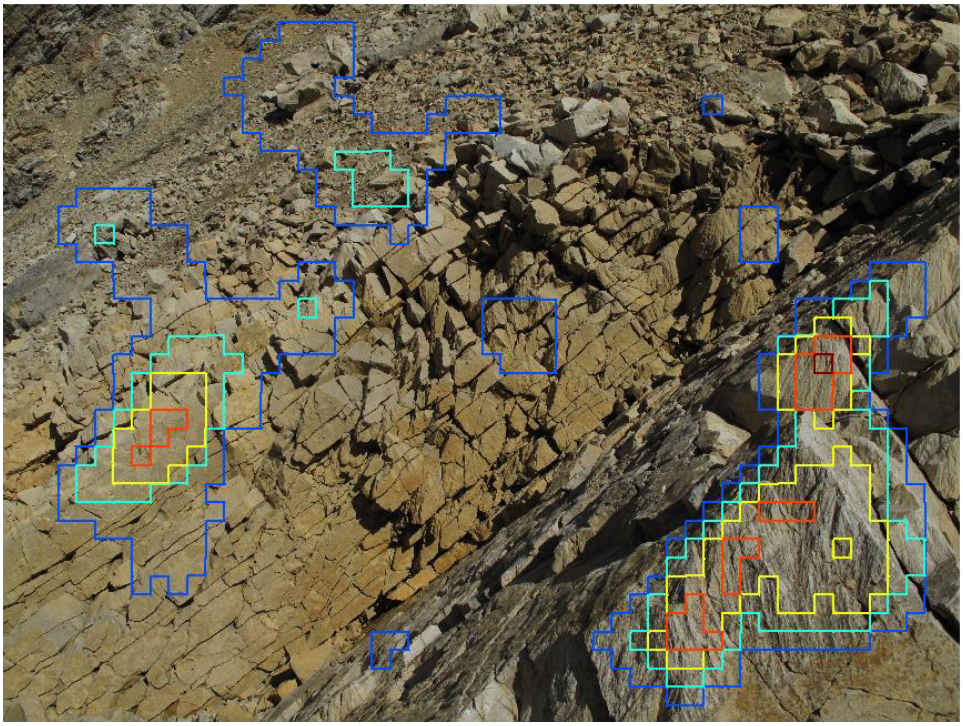


(a) Shatter-cone Probability at 24px

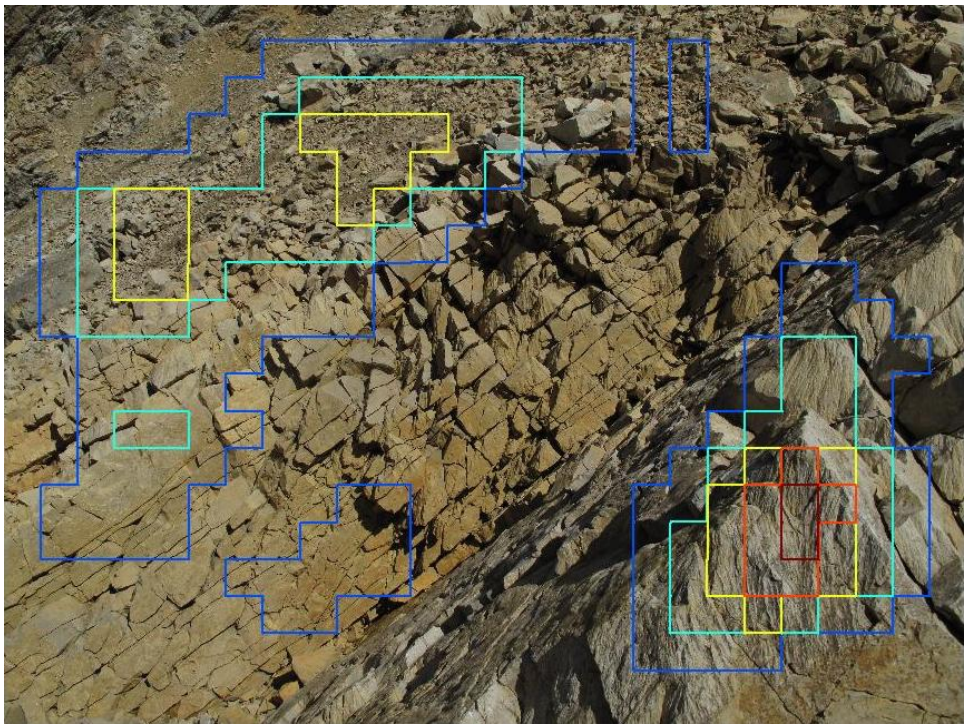


(b) Shatter-cone Probability at 100px

Figure 7.17: Shatter-cone Labelling based for various sub window sizes for the same image. The smaller size windows show better resolution in identifying small as well as large shatter-cones but comes at a cost of an exponential growth in processing times.

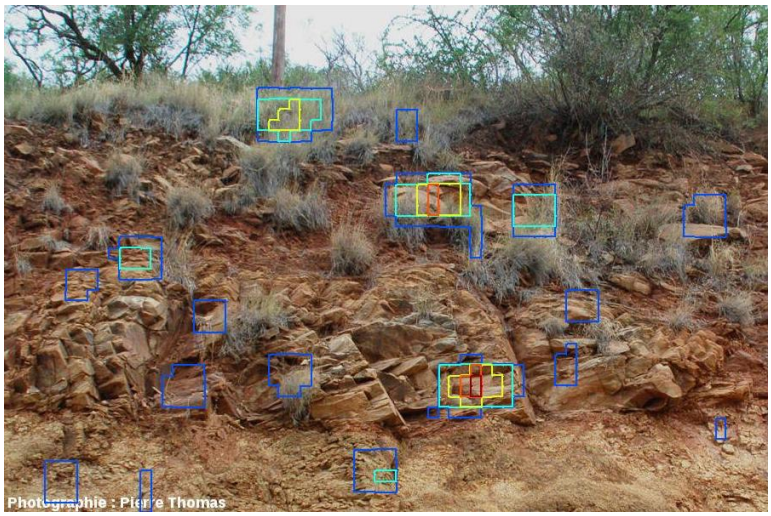


(a) Shatter-cone Probability at 200px

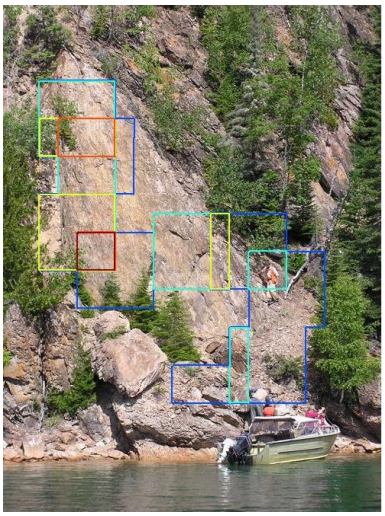


(b) Shatter-cone Probability at 400px

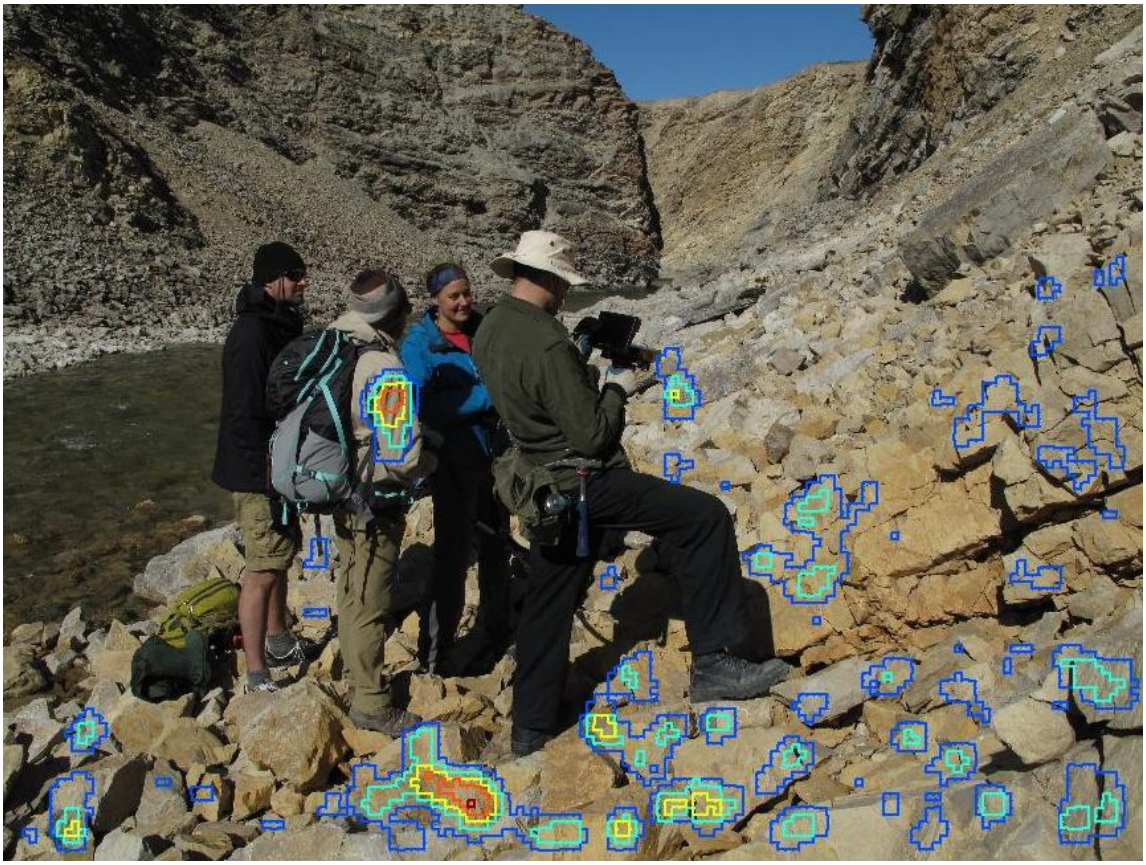
Figure 7.18: Shatter-cone Labelling based for various sub window sizes for the same image. The larger window size still detect larger scale shatter-cones, but as the window size increases smaller features are lost and error creeps into textured regions due to a loss of resolution.



(a) Vredefort Dome South Africa



(b) Slate Islands



(c) Prince Albert Impact Crater, Victoria Island NWT

Figure 7.19: A probability map of shatter-cones in novel landscape scale image

Chapter 8

Discussion, Conclusions and Future Work

The original conception of our work is predicated on three realities of robotic space exploration: First, that due to power, logistic, and temporal considerations the amount of data transferred between a spacecraft and earth is limited to a fraction of what a robotic system operationally might, and in fact does collect for situational awareness. Second, that there may exist in the moment to moment imaging of a robotic rover's images scientifically relevant information that is currently unexamined and not kept for later retrieval. And third, that there is an intrinsic value to a trainable subsystem which would be able to detect, with a reasonable certainty, features which may be interesting to a terrestrial researcher.

These realities mean that a discussion of the validity of our methods will be different from those of a vision system operating within the purview of a human operator, or those designed to replace a human operator for some purpose such as navigation. While most vision systems seek to maximize their accuracy and completeness while finding a balance between the occurrence of false positive and false negative labelling errors in order to remove a human decision maker from the process, our system is different in that we are seeking to bring a human decision maker into what otherwise would not be part of the data set available. This means that while accuracy is an important function of our work, completeness is less of a focus, and our system must minimize false positives if it is not to present an undue burden on the robot.

8.1 Outcrops

The identification of outcrops presents us with a unique set of challenges. The human identification of an outcrop is one that is very often context dependant. While we can make some generalities about the structure, makeup and location of an outcrop, a globally applicable method to identify all such features is almost certainly an unfeasibly hard problem given computing power required to achieve such a feat. The International Mineral Association has Identified 4985 different minerals [1]. Any given outcrop can be made up of an almost limitless combination of these minerals in varying amounts, differences that will alter the texture, colour, albedo and shape of that outcrop. Outcrops themselves can range from homogeneous cliffs, to an amalgam of rocks suspended within a binding material, to layer upon layer of material laid down over centuries. It is with this in mind that we must examine the success of our system. We never set out to create one monolithic system, but rather one that could function within the

operational area of a rover, and one that could be updated as the features changed.

8.1.1 Conclusions

The tests we have performed show that a boosted decision committee can be an effective way to identify outcrops given a relatively small training set. We also believe that an iterative building of test sets would allow for a continual updating of search software as a device moves through an environment. Since the imagery and rover traversal through an environment happens regardless, a probabilistic algorithm could be integrated to help increase the likelihood of identifying regions of interest without interfering with existing exploration methodologies. Likewise by refining the uninteresting training database to be as close to marginal decision making we believe that a novel item could be identified so long as they are sufficiently differentiable from the makeup of uninteresting image patches.

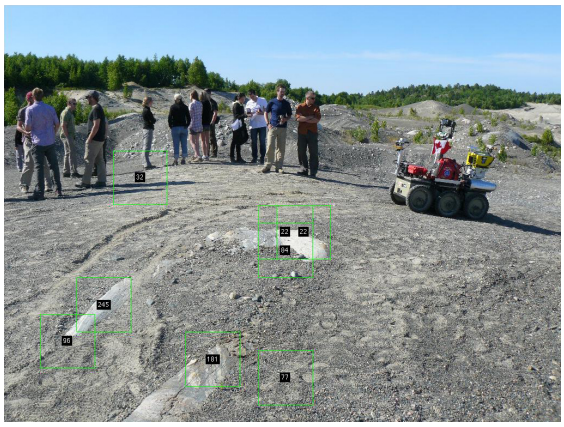
While most of our testing showed favourable outcomes there were a few specific cases which failed to offer useful information, and in fact offered completely erroneous results. The image identified in figure 7.9 did not simply over or under value the scene, but rather it identified a region which did not fit the intended parameters with a greater confidence than those regions which were manually identified as interesting. This failure however may not be systemic, but instead may be an artifact of the methodology we used to train our classifier, likewise, it is possible that such misidentifications could be minimized with a future iteration of our training data set. We will discuss this possibility in 8.1.2, along with an artifact of window size we noticed during our initial testing because we believe the two outcomes may be related.

Despite the relative success of our algorithms, it must also be noted that the methodology used is simply too computationally intensive for the processing and energy limitations of current spacecraft. Even terrestrial use of such algorithms would require a significantly powerful system to adequately identify objects of interests which would make realtime, or field use of this software more difficult. This does not mean that our approach is without merit. We also believe that there are many readily available ways to improve the computational requirements of this type of approach and we will discuss some of these in 8.1.2 and 8.4.

8.1.2 Discussion

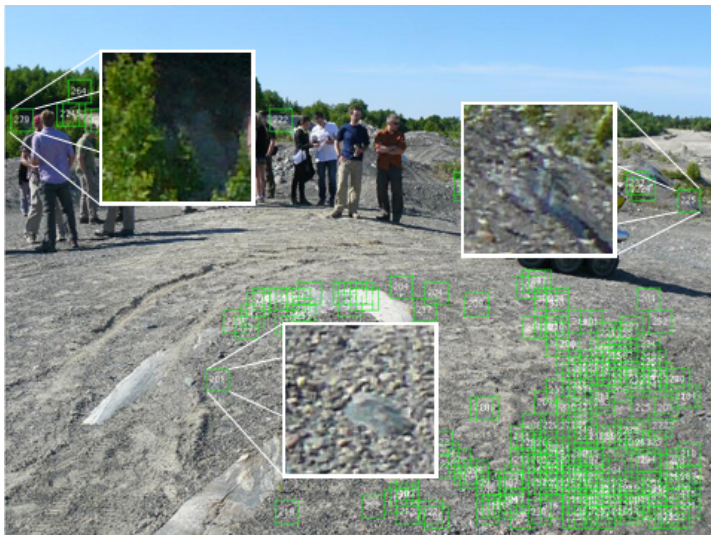
Detection Window Size

When we were performing our initial testing we noticed that image window size had a significant impact of the regions of interest identified by the decision making algorithm. As can be seen in the isolated positive classifications on the same image at different scales, the 100px window size in Figure 8.1b and the 300px window size in Figure 8.1a returned significantly different results. At first glance it appears that the 100px windows failed to pick up the obvious salient features, but upon closer examination in Figure 8.1c we find that at least some of the positive results, are genuinely correct positive results, or at least are reasonably indistinguishable from a positive result. While the probability map that was generated (seen in figure 7.8e) successfully identifies the foreground structures, the top two magnified squares in Figure 8.1c are outcrop structures located at distances which make them appear small enough such that at larger window sizes these features are overlooked.



(a) Results at 300px

(b) Results at 100px



(c) Close up of positive results

Figure 8.1: A comparison of areas of interest identified at different scales.

Likewise, if we compare the bottom magnified square in Figure 8.1c with a positive example from the training set such as the one in Figure 6.2a, we can see why our algorithm might conflate this sub window with a positive result. One possibility of reducing this error would be to isolate the examples in our positive training set to only include images of the outcrop itself, and not the surrounding materials. It is possible that this strategy would reduce some of the false positives we saw in Figure 7.9, but there are a few arguments to be made against this strategy. If we examine the positive identifications in larger scale windows, such as those seen in Figure 7.8f and Figure 8.1a we can see that most of those positive identifications also include some of the surrounding material. Lacking some form of image segmentation, it is unlikely that an image window would only contain exclusively rock from an outcrop. Such a strategy would also reduce the number of positive training examples that could be gathered from available photographs, meaning that a larger number of unique examples would be required to build a sufficiently large training database. To compensate, we could arguably use

a smaller sub-window, but this too is problematic because it would cause the number of sub-windows which needed to be processed to be processed to grow at an exponential rate over the same image resolution.

Portability of Outcrop Features

As we have discussed at the beginning of this section, it would be next to impossible to make a universal training set to cover all possible combinations of rock. Thankfully, there is no need to do so since the conditions which create rock formations on any large enough scale generally cover a large enough geographical area to allow for transitions from one rock type to the next. In those cases where this does not happen, there are often other such markers which would suggest a priori that such a radical shift might occur. The question for further work is: how much can a rock formation change before our descriptors become unreliable. While we did not go into great detail regarding this work, we do have a suggestion that some variability in rock constituency does not preclude detection.

8.2 Shatter-Cones

Shatter-cones are a unique geological feature, which tell a very specific tale of a very specific event in the history of a geological region. While they are not common, their identification would present a calling card of a transformative event to scientists exploring an alien landscape. To find such an object is to take a giant leap in the understanding of an impact event. So useful an indicator would a shatter-cone be that there is a value in having a rover ”keeping an eye” out for one as it traverses an alien crater or impact basin.

8.2.1 Conclusions

Over the course of our four tests we have shown that our method offers a reasonable way of detecting shatter-cones within an image. Within all images that contained shatter-cones, we were able to identify at least one instance of a shatter-cone, and while our method is not comprehensive in its detection of all shatter-cone features, we found that in each image we tested containing shatter-cones, we were able to identify with a useful certainty at least part of a shatter-cone structure. Moreover by weighing the probability of a given region within a larger image to contain a shatter-cone we have shown that it is possible for an autonomous system to select a small subset of an image for conformation by secondary systems, either human or autonomous.

8.2.2 Discussion

Any discussion of results of classification must begin by noting that the variability found within the appearance of existing shatter-cones is quite large, with terrestrial examples ranging from the obvious to those which require an expert eye to detect. The positive examples contained within our training set only included those which most clearly demonstrated the features associated with shatter-cones. This means that our method has been designed from the beginning

to identify and label a partial, but not insignificant subset of shatter-cones. Over the course of testing it was found that the ability to identify shatter-cones extended beyond the examples used for our training data base as we will discuss below.

One final limitation to the experimental results obtained should be brought up before proceeding. In the course of identifying images for testing it became immediately clear, that there is an under representation of shatter-cones in situ where the shatter-cones do not make up a large portion of the image. Most of the imagery identified as containing shatter-cones are photos taken in the field that were taken to highlight the cones themselves. This meant that in most test images, shatter-cones either make up a substantial percentage of the rocks in the image or are not present at all. While in theory the methodology we used should be able to successfully identify shatter-cones in relatively small regions of an overall image we were unable to test this with enough examples to be completely confident in our results despite the success we experienced.

8.3 Boosting as a Strategy in Extraterrestrial Environments

While we were able to achieve mostly positive outcomes with this work, ultimately the question that must be addressed is: Does it make sense to use this approach in the robotic exploration of extraterrestrial environments? If the answer to this question is yes, then the question becomes: What are the practical considerations required to make it happen?

8.3.1 Validity of this Approach

As the results of our work has shown, there is reason to believe that a system designed for the environment in which the rover is exploring offers a reasonable chance of being able to distinguish an object of scientific interest from uninteresting ones using the boosted committee approach. Since much of what we are looking for is currently either offloaded to human controllers or is not examined at all even a modest ability represents a positive step in the abilities of an autonomous system. Likewise such a system offers a great potential for shifting information processing loads away from systemic bottle necks when it achieves a high enough level of accuracy and computational efficiency.

Methodological Benefits

The approach chosen was specifically selected for the purpose of minimizing the data load, on the more tenuous elements of the deep space network and rover capabilities and offers a number of compelling advantages to its use by an extraterrestrial rover.

Terestrial Training The most computationally complex part of the process the development of the decision making committee can be handled by terrestrial computers and decision making committees can be constantly improved in an iterative fashion as data sets and methodologies improve.

Object Type Scalability Most if not all of the image features that make up image descriptors are universal enough to be applied to a number of geological or meteorological features that might be of interest. Once a vocabulary of image features is created from an image the ability to identify different features using different combinations of them means that they could be used on many different targets.

Data Availability Since vision systems make up the most significant portion of current rovers navigation system, much of the imagery is a byproduct of robot traversal and does not require the addition of complex instrumentation or a cessation of travel in order to perform it's primary identification.

Multi-instrument Collaboration The ability to train a feature set need not be limited to simply a single instrument. Since boosted comities work on a data vector of descriptors, that data can come from more than a single visible spectrum camera. Once calibrated to account for image registration, stereo image features could be added to a data vector, as well as different spectra such as IR and UV, Likewise the byproducts of other navigational techniques such as LIDAR point clouds could also be incorporated.

Methodological Limitations

While conceptually the method used has a number of benefits, it is also not without drawbacks. There are a number of factors which might limit the use of such an approach in the real world.

Computational Complexity As features are added the size of the data vector increases, which means that the time to create the feature vector increases. With the vector size and features we chose, the creation of a feature vector of a 100×100px sub-window created a vector F where $F_{size} = 7281$ (see table 7.2.) This required a processing time of about 1 second for a relatively modern dual core intel processor. While this is untenable for existing processing power, it is not inherent to the methodology, and with different feature methodology it is within the forecasted processing abilities of upcoming rovers.

Resolution vs. Image Size As our experiments showed the larger window size we used limited the resolution we were able to achieve in detection, while the smaller the size the greater the number of sub-windows needed, and this number grew at an exponential rate. To achieve the resolution achieved in the image shown in figure 7.17a almost 100,000 sub-windows were analyzed.

8.4 Future Work

8.4.1 Methodological Improvements

Expanded Detection Vocabulary While we selected a number of useful features for extraction, our list is far from comprehensive. Similar work performed by others, has identified a number of comparable methods of creating a vocabulary of image features which may in

combination improve the overall efficiency of our approach. (See [24] and others) a more systematic approach to this problem may be of some benefit.

Programmatic Efficiencies In order to improve the resolution of detection while minimizing the computational complexity a number of strategies might be employed such as image pyramids, segmentation etc. Likewise the code we utilized was written in a high level language(MATLAB) and could benefit from a more efficient language, as well as more efficient methods to eliminate reduplication.

8.4.2 Structural Improvements

Data Set Creation One of the most challenging aspects of this work is finding a dataset of labeled images for training and testing. NASA image databases are generally organized geographically and temporally and not well labeled by features or geography (there are some rare exceptions) The classification of geological images is generally not standardized and to a certain degree dependant on subjective measures and descriptions.

Application Specific Hardware Cameras are designed for the human eye, and as such deal with and store data in a fashion that is conducive to human vision. The use of software for image filters has some inherent inefficiencies which, while almost unnoticeable to modern computers, could be problematic for the limited resources of a rover. Some of the more common filtering could be performed through hardware filters such as FPGAs or specifically designed parallel processors.

8.5 Final Remarks

The algorithms developed here have been shown to perform their work well. Properly designed and implemented, a version of this work can offer flexibility to autonomous systems in performing science in the absence of a human presence. While this method currently requires a computational power that is not available on the present generations of rovers, the refinement of these techniques combined with increased computational power and efficiency will allow these kinds of techniques to be utilized by a future generation of robotic explorers.

Bibliography

[1] International Mineral Association. The new international mineral association list of minerals. [Online; Last accessed 28-March-2015].

[2] Timothy Barfoot, Paul Furgale, Braden Stenning, Patrick Carle, Laura Thomson, Gordon Osinski, Michael Daly, and Nadeem Ghafoor. Field testing of a rover guidance, navigation, and control architecture to support a ground-ice prospecting mission to mars. *Robotics and Autonomous Systems*, 59(6):472–488, 2011.

[3] Jeffrey J Biesiadecki and Mark W Maimone. The mars exploration rover surface mobility flight software driving ambition. In *Aerospace Conference, 2006 IEEE*, pages 15–pp. IEEE, 2006.

[4] Elisabeth T Bowman, Kenichi Soga, and W Drummond. Particle shape characterisation using fourier descriptor analysis. *Geotechnique*, 51(6):545–554, 2001.

[5] Matthew Brown and David G. Lowe. Invariant features from interest point groups. In Paul L. Rosin and A. David Marshall, editors, *BMVC*, pages 1–10. British Machine Vision Association, 2002.

[6] J. S Bruner, J. J. Goodnow, and G. A. Austin. *A study of thinking*. John Wiley, New York, 1956.

[7] H.P.V. C. Method and means for recognizing complex patterns, December 18 1962. US Patent 3,069,654.

[8] R Castano, RC Anderson, J Fox, JM Dohm, AFC Haldemann, and W Fink. Automating shape analysis of rocks on mars. In *Lunar and Planetary Science Conference*, volume 33, page 2000, 2002.

[9] R. Castano, T. Estlin, D. Gaines, C. Chouinard, B. Bomstein, R.C. Anderson, M. Burl, D. Thompson, A. Castano, and M. Judd. Onboard autonomous rover science. In *Aerospace Conference, 2007 IEEE*, pages 1 –13, march 2007.

[10] R. Castano, M. Judd, T. Estlin, R.C. Anderson, D. Gaines, A. Castano, B. Bornstein, T. Stough, and K. Wagstaff. Current results from a rover science data analysis system. In *Aerospace Conference, 2005 IEEE*, pages 356 –365, march 2005.

[11] Rebecca Castano, Robert C Anderson, Tara Estlin, Dennis DeCoste, Forest Fisher, Daniel Gaines, Dominic Mazzoni, and Michele Judd. *Rover traverse science for increased mission science return*. Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2003.

[12] Rebecca Castano, Tara Estlin, Robert C Anderson, Daniel M Gaines, Andres Castano, Benjamin Bornstein, Caroline Chouinard, and Michele Judd. Oasis: Onboard autonomous science investigation system for opportunistic rover science. *Journal of Field Robotics*, 24(5):379–397, 2007.

[13] Rebecca Castano, Tobias Mann, and Eric Mjolsness. Texture analysis for mars rover images. In *SPIE's International Symposium on Optical Science, Engineering, and Instrumentation*, pages 162–173. International Society for Optics and Photonics, 1999.

[14] Bidyut Baran Chaudhuri and Nirupam Sarkar. Texture segmentation using fractal dimension. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(1):72–77, 1995.

[15] William R. Church. Geological history of the southern province of ontario.

[16] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[17] R.C. Crida and G. De Jager. Rock recognition using feature classification. In *Communications and Signal Processing, 1994. COMSIG-94., Proceedings of the 1994 IEEE South African Symposium on*, pages 152–157, Oct 1994.

[18] R.C. Crida and G. de Jager. Multiscalar rock recognition using active vision. In *Image Processing, 1996. Proceedings., International Conference on*, volume 1, pages 345–348 vol.2, Sep 1996.

[19] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR (1)*, pages 886–893. IEEE Computer Society, 2005.

[20] Daniel C Dennett. Quining qualia. *Consciousness in modern science*, 1988.

[21] Oxford English Dictionary. "rock, n.1", January 2012.

[22] Gather R Drevin. Using entropy to determine the roundness of rock particles. In *Signal Processing Proceedings, 2000. WCCC-ICSP 2000. 5th International Conference on*, volume 2, pages 1399–1404. IEEE, 2000.

[23] R.O. Duda and P.E. Hart. Use of the hough transformation to detect lines and curves in pictures. Technical Report 36, AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025, Apr 1971. SRI Project 8259 Comm. ACM, Vol 15, No. 1.

[24] Heather Dunlop. Automatic Rock Detection and Classification in Natural Scenes. Master's thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, September 2006.

[25] DJ Durian, H Bideaud, P Duringer, A Schroder, and CM Marques. The shape and erosion of pebbles. *arXiv preprint cond-mat/0607122*, 2006.

[26] Dr. Kord Ernstson and Fernando Claudin. Ernston claudin impact structures: The shatter cone page.

[27] Tara A. Estlin, Benjamin J. Bornstein, Daniel M. Gaines, Robert C. Anderson, David R Thompson, Michael Burl, Rebecca Castano, and Michele Judd. AEGIS Automated Science Targeting for the MER Opportunity Rover. *ACM Transactions on Intelligent Systems and Technology*, 3(3):1–19, June 2012.

[28] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[29] William T. Freeman and Michal Roth. Orientation histograms for hand gesture recognition. Technical Report TR94-03, MERL - Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, December 1994.

[30] Bevan M French. Shock metamorphism of natural materials. *Science*, 153(3738):903–906, 1966.

[31] Bevan M French. Traces of catastrophe: A handbook of shock-metamorphic effects in terrestrial meteorite impact structures. *Technical Report, LPI-Contrib-954*, 1, 1998.

[32] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In Paul M. B. Vitányi, editor, *EuroCOLT*, volume 904 of *Lecture Notes in Computer Science*, pages 23–37. Springer, 1995.

[33] Martha S. Gilmore, Rebecca Castao, Tobias Mann, Robert C. Anderson, Eric D. Mjolsness, Roberto Manduchi, and R. Stephen Saunders. Strategies for autonomous rovers at mars. *JOURNAL OF GEOPHYSICAL RESEARCH*, 105(12):29–223, 2000.

[34] Xiaojin Gong and Jilin Liu. Rock detection via superpixel graph cuts. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 2149–2152. IEEE, 2012.

[35] V Gor, R Castano, R Manduchi, RC Anderson, and E Mjolsness. Autonomous rock detection for mars terrain. *Space*, pages 1–14, 2001.

[36] James P Hyslip and Luis E Vallejo. Fractal analysis of the roughness and size distribution of granular materials. *Engineering Geology*, 48(3):231–244, 1997.

[37] L Lepisto, Iivari Kunttu, Jorma Autio, and Ari Visa. Comparison of some content-based image retrieval systems with rock texture images. In *Proceedings of 10th Finnish Artificial Intelligence Conference, Oulu, Finland*, pages 156–163. Citeseer, 2002.

[38] Leena Lepisto, Iivari Kunttu, Jorma Autio, and Ari Visa. Classification method for colored natural textures using gabor filtering. In *Image Analysis and Processing, 2003. Proceedings. 12th International Conference on*, pages 397–401. IEEE, 2003.

[39] Leena Lepistö, Iivari Kunttu, and Ari Visa. Rock image classification using color features in gabor space. *Journal of Electronic Imaging*, 14(4):040503, 2005.

[40] Scott Lever, Mike Seibert, and Al Herrera. sols 3017-3022, july 19-24, 2012: Opportunity picks up the pace with several drives this week, July 2012.

[41] Rongxing Li, Kaichang Di, Andrew B Howard, Larry Matthies, Jue Wang, and Sanchit Agarwal. Rock modeling and matching for autonomous long-range mars rover localization. *Journal of Field Robotics*, 24(3):187–203, 2007.

[42] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision, Corfu*, 1999.

[43] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[44] Larry Henry Matthies. *Dynamic Stereo Vision*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1989. AAI9023429.

[45] Tom Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[46] Simphiwe Mkwelo, Frederick Nicolls, and Gerhard De Jager. Range and intensity vision for rock-scene segmentation. In *Progress in Pattern Recognition, Image Analysis and Applications*, pages 340–349. Springer, 2007.

[47] Robert J Oakey, Michael Green, Paul A Carling, Mark WE Lee, David A Sear, and Jeff Warburton. Grain-shape analysis—a new method for determining representative particle shapes for populations of natural grains. *Journal of Sedimentary Research*, 75(6):1065–1073, 2005.

[48] Clark F Olson and Larry H Matthies. Maximum likelihood rover localization by matching range maps. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 1, pages 272–277. IEEE, 1998.

[49] Clark F Olson, Larry H Matthies, Marcel Schoppers, and Mark W Maimone. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4):215–229, 2003.

[50] Ives Rey Otero and Mauricio Delbracio. The Anatomy of the SIFT Method. *Submitted (Image Processing On Line preprint)*, pages 1–28, 2012.

[51] Mari Partio, Bogdan Cramariuc, Moncef Gabbouj, and Ari Visa. Rock texture retrieval using gray level co-occurrence matrix. In *Proc. of 5th Nordic Signal Processing Symposium*, volume 75. Citeseer, 2002.

[52] PASSC. Earth impact database.

[53] Markus Peura and Jukka Iivarinen. Efficiency of simple shape descriptors. In *Proceedings of the third international workshop on visual form*, volume 443, page 451. Citeseer, 1997.

[54] Ioannis Rekleitis, J-L Bedwani, and Erick Dupuis. Autonomous planetary exploration using lidar data. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3025–3030. IEEE, 2009.

[55] David John Roddy, Robert Osborne Pepin, and Russell Blair Merrill. Impact and explosion cratering: Planetary and terrestrial implications; proceedings of the symposium on planetary cratering mechanics, flagstaff, ariz., september 13-17, 1976. In *Impact and Explosion Cratering: Planetary and Terrestrial Implications*, volume 1, 1977.

[56] Amir Sagy, Ze'ev Reches, and Jay Fineberg. Dynamic fracture by large extraterrestrial impacts as the origin of shatter cones. *Nature*, 418(6895):310–313, 2002.

[57] Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[58] David R Thompson, Trey Smith, and David Wettergreen. Data mining during rover traverse: From images to geologic signatures. *iSAIRAS, Münich, Germany*, 2005.

[59] DR Thompson, S Niekum, T Smith, and D Wettergreen. Automatic detection and classification of geological features of interest. In *Proceedings of the IEEE Aerospace Conference*, 2005.

[60] Sebastian Thrun. What we're driving at, Oct 2009.

[61] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9):661–692, 2006.

[62] Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

[63] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511 – I–518 vol.1, 2001.

[64] Richard Volpe, Todd Litwin, and Larry Matthies. Mobile robot localization by remote viewing of a colored cylinder. In *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 1, pages 257–263. IEEE, 1995.

[65] Li Wang. Automatic identification of rocks in thin sections using texture analysis. *Mathematical geology*, 27(7):847–865, 1995.

Curriculum Vitae

Name:	Greg Elfers
Post-Secondary Education and Degrees:	University of Western Ontario London, ON 2011 - 2015 M.Sc.
Honours and Awards:	NoAE INNOVATION AWARD (RoadLAB) 2010
Related Work Experience:	Teaching Assistant The University of Western Ontario 2011 - 2012 Research Assistant Western Engineering 2013 - 2015 Embedded Systems Programmer EcoInsight Instruments 2014 - 2015