Autonomous Mobile Robot Localization in Large-Scale Environments Using Only a Camera

by

Eric L. Akers

University of Kansas, 2007

B.S. Computer Science, 2001
M.S. Computer Science, 2003

Submitted to the graduate degree program in Electrical Engineering and Computer Science and the Faculty of the Graduate School of the University of Kansas In partial fulfillment of the requirements for the degree of Doctor of Philosophy

_____
Dr. Arvin Agah (Chairperson)

_____
Dr. Prasad Gogineni

_____
Dr. Donna Haverkamp

_____
Dr. Carl Leuschen

_____
Dr. David Braaten

Date Defended: _____

The Dissertation Committee for Eric L. Akers

certifies that this is the approved version of the following dissertation:

Autonomous Mobile Robot Localization in Large-Scale Environments Using Only a Camera

Committee:

_____

Dr. Arvin Agah (Chairperson)

_____

Dr. Prasad Gogineni

_____

Dr. Donna Haverkamp

_____

Dr. Carl Leuschen

_____

Dr. David Braaten

Date Approved: _____

Abstract

Localization is a fundamental problem of autonomous mobile robots. Localization is the determination of the position and orientation of a robot. Most localization systems are made up of several sensors and a map of the environment.

The problem of localization can be approached in different ways based on the environment within which the robot operates. The environment and conditions can affect which sensors are used. For example, robots that operate indoors cannot use GPS, and outdoor robots may not be able to utilize sonar well in wide open spaces.

Sophisticated localization systems can solve both the global location problem and the kidnapped robot problem. Global localization is the process of placing the robot into an unknown location within the map, and the robot should be able to locate itself within a relatively short period of time. The kidnapped robot problem is similar to global localization, as it is a test of how well the robot is able to recover after becoming lost. The robot is "teleported" to a new location, and the robot should again be able to determine its new location within a relatively short amount of time.

The PRISM/CReSIS project is developing autonomous robots in an effort to measure ice sheets characteristics in Greenland and Antarctica. These robots currently rely on differential GPS for localization and navigation. In order to

survive for long periods of time in these environments, however, the robots needs to be able to return to camp sites in order to refuel and unload the data that has been acquired. In order to perform this task effectively and safely, a more elaborate system is required. A localization system that can recognize the different locations of the camp sites is the beginning of this process.

The approach of this dissertation is to use a single camera for use in multiple types of large-scale environments: indoors, outdoors, and in polar camp sites in Greenland and Antarctica. The camera is selected as the sensor for several reasons. First, a single image potentially contains a lot of information that can be used in many different ways. Second, the size of a camera allows for the system to be used on many different platforms, including those with limited payloads such as UAVs (uncrewed aerial vehicles). This also allows for the system to be very portable if necessary, and can be plugged into already existing systems more easily. Lastly, the cost of cameras allow for the system to be used in large quantities. For example, a potential application of this system is using teams of robots for seismic sensing. This would require many cameras for use on numerous mobile robots.

In order to work in large-scale environments, a hybrid map approach has been used. The hybrid map includes both topological and geometric maps. The topological maps allow for the system to scale more easily, and the geometric maps allow for the system to localize to a finer scale. These are performed in a two step approach. First, a general location is decided based on topological localization. Then the geometric localization step is performed using the geometric step stored in the location of the topological map, which is selected in the first step.

The system described in this dissertation uses an appearance-based approach for recognizing the different locations. The appearance-based methods attempt to

recognize the appearance of a scene rather than specific objects. Several different types of features are tested including histograms, eigenimages, and Hu Moments for topological localization, and SIFT for geometric localization.

For the topological map, the features are modeled using a Gaussian mixture model. Then a hidden Markov model is trained as the classifier. Only the recognition of images is necessary for determining location. The geometric localization relies on the matching of SIFT keypoints for determining the position and orientation on the geometric maps.

The metrics for the topological testing include the percentage of images that are localized correctly, the number of images required to perform global localization, and the number of images required to solve the kidnapped robot problem. The measurement is based on number of images rather than actual time because all testing occurs offline, and images can be captured at different rates.

The results of the testing showed that 95% of the images for non-polar regions, both indoor and outdoor, were localized correctly with respect to the topological map. The system typically required at least two images to solve the global localization problem, and around three images to solve the kidnapped robot problem.

Topological testing was also performed using images from polar camp locations, but the results are inconclusive because of the relatively few number of images. The system is able to localize 20% of the images.

The metrics for the geometric testing are position accuracy and orientation accuracy. Position accuracy is the percentage of images that are correctly localized with respect to the position. Orientation accuracy is the percentage of images that are correctly localized with respect to the orientation.

The geometric testing is performed in non-polar locations, both indoors and

outdoors. The experiments result in 94% of the images being localized correctly for the position, and 90% of the images for the orientation. These images are localized to within 1 foot and 45 degrees of the actual position and orientation.

More testing will have to be performed to achieve better results in the polar camp sites, as well as using SLAM methods to automatically build the geometric maps because of the amount of time required to do this manually. This approach can be extended to work with the PRISM/CReSIS polar robots to allow for automated fueling and unloading of data.

Acknowledgment

I would first like to thank my Wife, Vicki, for her support and patience during my many years as a student. I would also like to thank my advisor, Arvin Agah, for his help and guidance throughout my career as a graduate student.

I am extremely grateful to Dr. Gogineni for the support of my research and also for the opportunities to be involved with and work on PRISM and CReSIS. My work has been sponsored by PRISM and CReSIS since June 2000, including field experiments in Greenland and Antarctica.

Next, I would like to thank my committee for their guidance, and Dr. Jim Miller for acting as my teaching mentor.

I would also like to thank my parents and my in-laws for all of their help, especially in the last several years. I appreciate all the support from the entire family. I am also grateful to my nephews, Logan and Ethan, for providing the entertainment that only 2.5 and 1.5 year olds can.

Leon Searl deserves a great deal of thanks for his friendship and mentorship during my early years as a research assistant.

To all my colleagues at PRISM and CReSIS, especially Richard Stansbury, Chris Gifford, and Hans Harmon, I have enjoyed my time working with you, and I hope to continue to be able to work with the center in my future endeavors.

This work was supported by the National Science Foundation (grant #OPP-

0122520), the National Aeronautics and Space Administration (grants #NAG5-12659 and NAG5-12980), the Kansas Technology Enterprise Corporation, and the University of Kansas.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Localization is a fundamental problem that must be addressed in autonomous mobile robotics. Localization is defined as determining the *pose*, position and orientation, of the robot from the sensor data, given a map of the environment. It is usually specified as the 3D vector (x, y, θ) where x is the x position, y is the y position, and θ is the orientation.

Most localization systems use a probabilistic method of localization using Bayes' Rule to compute the posterior $P(l \mid s, m)$ where l is the location, s is the sensor data, and m is the map. This is the probability of being at location l given sensor data s and a map of the world, m. The probabilistic methods are described in a later section.

The map of the environment can be created before hand, or the robot may build its map as it moves. The latter systems are usually described as SLAM, Simultaneous Localization and Mapping. Much work has gone into these types of systems more recently. Some argue that a mobile robot cannot truly be autonomous without being able to move to previously unexplored environments. These types of systems can be used for search and rescue robots as well as service

robots.

Some robots however, require a map to be able to perform their tasks. Soccer robots and delivery robots are examples of robots that cannot work without an a priori map. Some systems use an exploration routine from their robot in order to build the map. The exploration method usually only works indoors as it can be difficult to set boundaries for the exploring robot in outdoor environments.

The required accuracy of the localization system depends on the application domain of the system. A soccer playing robot may require less accuracy than a robot whose job is to wash cars because of the possibility of damaging the cars.

The PRISM/CReSIS [21] polar robot named Marvin [1] [3] [5] [24] [59] [60] uses DGPS to localize. While the robot does not have an actual map, it uses the GPS coordinates intrinsically as a hybrid topological and geometric map. The hybrid maps are described in more detail in section 2.1.5. The GPS coordinates can be accurate to within a few centimeters depending on the distance to the base-station of DGPS. The orientation of the robot can be determined by generating a vector from the current coordinate and the previous coordinate.

The PRISM/CReSIS project's goal is to eventually have multiple autonomous robots working at the same time. In the future, a distributed localization system can be designed where each robot localizes itself with the help of the others. A system using this method might have enough accuracy to be able to use a regular GPS instead of relying on the costly DGPS.

Many camps exist in Antarctica and Greenland for the purpose of research, drilling, or other activities. Any of these could be used as a base camp for the polar robot to measure the ice sheets from. These camps usually have areas designated for sleeping, areas designated for gathering snow used for the camp water supply, and other areas that are dangerous to go through such as a drilling site.

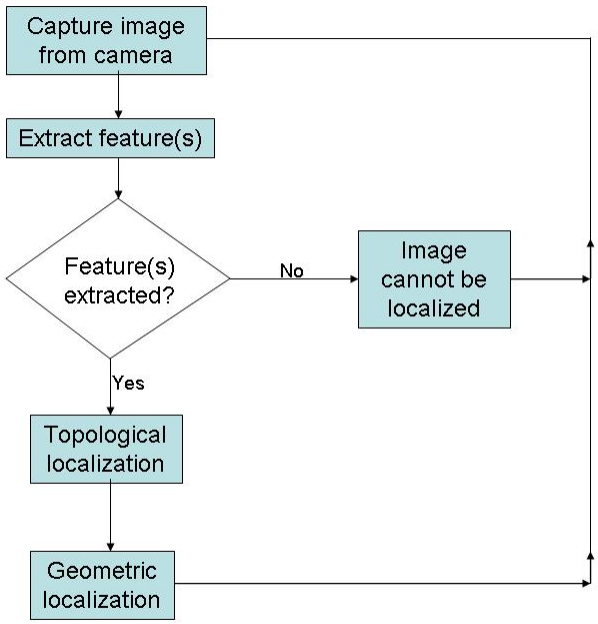

Figure 1.1: Flow chart showing the decision process of the localization system

The polar robot will eventually be completely autonomous and have to decide when it needs to go back to the camp. Therefore, a less generic localization system would be helpful in this situation when it must be able to move to a specific location within the camp. The robot should be able to make it to a garage to refuel or get repairs without having to drive through restricted areas. There will most likely be specific locations in a polar camp where the robot should enter and exit from that will require a more advanced localization system.

1.1 Problem Statement

A localization system that works both indoors and outdoors using only a camera can be used in large scale environments. "Large-scale environments" are defined as "if its spatial structure is at a significantly larger scale than the sensory horizon

of the observer [34]."

The approach taken is to base the decision on a two step process shown in Figure 1.1. The two steps are described as toplogical localization and geometric localization. These are each describe in detail in later chapters.

1.2 Motivation

This dissertation describes the design, development, and evaluation of a localization system that can work in several different environments using only a camera. This type of a system has the advantage of being able to use a simple and relatively inexpensive sensor that is easily portable. Because of the simplicity of the system, it can be used in combination with other systems to improve the system as a whole.

Using only a camera for localization allows for a much smaller system than a typical system that uses stereo cameras or a laser range finder, or any combination of sensors that might be bulky. Using only a camera also allows the system to be moved from one vehicle to another. For example, a person that requires a wheel chair may move the system from the car to the wheel chair.

The single camera system allows for it to be used in very small devices like a PDA or a phone. The system will allow for a portable system that can work over a large area, whether it is used to guide a human or a robot.

1.2.1 PRISM/CReSIS Project

The Polar Radar for Ice Sheet Measurement project is currently underway at The University of Kansas [33]. It is part of the Center for Remote Sensing of Ice Sheets (CReSIS) [15]. This project's goal is to develop radar systems to measure polar ice sheet properties in order to accurately determine their mass balance. This

Figure 1.2: Image of the PRISM/CReSIS Robot Marvin II in Antarctica

data will help researchers determine the contributions of polar ice sheet melting to global climate change and its effects on the rising sea levels.

Different radar systems have been developed for this task [33]. In order to accommodate these radar systems, an autonomous mobile robot, Figure 1.2, is being developed to tow the radar equipment over a large area. After the robot completes its traversal, it needs to return to camp to refuel and unload the data.

For the traversal over the ice sheets, the robot utilizes DGPS for navigation. However, once the robot returns to camp, it is desirable for the vehicle to utilize other sensors for navigation as well in order to drive through the camp safely and accurately. A system like the one described in this dissertation will allow the robot to safely find the fuel station and the location for unloading the data without driving through places such as tent city, the snow gathering site for the

camp water supply, or other areas that may be off limits.

If the tasks of fueling and unloading the data can be automated as well, then the time to get the robot out of camp and performing data collection again can be greatly reduced and require less human assistance.

PRISM/CReSIS is also looking towards using UAVs (uncrewed aerial vehicles) to take radar measurements. A localization system that relies on a camera could work very well for the UAVs because of their limited carrying capacity. GPS can sometimes give errors within 300ft which is significant when trying to land on a runway. Adding a camera to the system could help the UAV locate and stay within the runway when landing as well as keep the vehicle on track when taking off.

It is shown that multiple robots in precise formations can be used to record seismic activity in the polar regions [20]. The robots have been simulated usually using range sensors. Sonar has limited range and the data can be very noisy. Laser range finders can be bulky and heavy as well as increasing the cost of each individual robot by several thousand dollars. Both of these problems can be difficult to overcome. A solution could be to use a system based on cameras in combination with sonar and laser range finders.

Challenges of Polar Robots Polar robots have more challenges to overcome than typical indoor or outdoor robots. Polar robots must deal with harsh terrain, low visibility, and extremely low temperatures. This causes many problems for the typical sensor suite used on robots today. These are detailed more in [4].

Odometry is known to produce errors in environments that are not as difficult to traverse as polar areas. Odometry therefore, cannot be relied on to produce reliable results, even for short distances because of slippage. Inertial devices such

as gyros can be used to determine motion, but this can be difficult and error accumulations due to drift can make these values unreliable as well.

Cameras must be heated so that fog and ice do not form on the camera. Because of blizzards and other times when visibility is low, cameras cannot be depended on to operate at all times so other sensors must be used to make the system dependable. The terrain does not have much texture making it difficult to distinguish between different locations.

Laser range finders have similar problems in that it is difficult to distinguish different locations by the wide-open terrain. Blizzards can also cause range finders to give erroneous values as well.

GPS is the most reliable method of determining location. Because of solar activity and extreme low temperatures, GPS cannot be completely relied on for continuous operation.

The solution to dealing with these problems is to use a system that relies on multiple sensors in order to make the robot more robust. Odometry, gyros, range, and vision can be used together to determine relative motion of the robot. For example, a camera, rate gyro, and accelerometer are used simultaneously in [61] to determine vehicle motion.

In conditions where all sensors give erroneous data, the only solution may be to stop the robot until conditions are acceptable again.

1.2.2 Applications

This section describes some of the potential applications of a localization system of the type presented in this dissertation.

Package Delivery Delivery of packages by either a service robot or a human can benefit from a localization system. Truck drivers already use GPS systems to direct them to their next location. Assuming the GPS system has enough accuracy and an accurate map, it could be used in an area such as an industrial park or a college campus where there are many buildings and possible locations for delivery.

For delivering packages to specific locations within buildings such as a college campus mail system, a more complex localization could be used to direct the delivery person to the specific location using a PDA with a camera or automated delivery robots could be used.

Tourism A PDA with a camera could be used as a tour guide for locations like the Smithsonian or Disneyland. A map of the area could show the current location and orientation of the individuals on the tour as well as other interesting locations and provide information about them.

Service Robots Many robots that are built are meant to be service robots. Service robots provide assistance to humans in many different ways. Wheel chair service robots have been popular such as the Bremen Autonomous Wheelchair [36].

Wheel chair service robots, autonomous or not, can benefit from having a lightweight localization system. Assuming that a PDA with a camera or a laptop and a camera can be mounted on the wheel chair, the proposed system could be used in known environments. For example, an autonomous wheel chair could be instructed to go to a specific building on a college campus from any location on campus. The person can go to his or her next class with little effort.

Self Driving Automobiles Self driving vehicles have been tested recently in the DARPA Grand Challenge [11] and have been featured in many sci-fi movies. These vehicles could be used in industry or by the average commuter. For example, a self driving truck could be used to haul salt out of a salt mine. Some of these large trucks are expensive so that they must be run 24 hours a day in order to recoup the cost of the vehicle. An automated driving system could help improve the efficiency.

These vehicles will most likely rely on GPS for long distance traveling, but will not work well in areas where the GPS signal is low or in construction or mining sites such as a salt mine. Therefore, another system will have to be used to help localize the vehicles for everyday use.

1.3 Dissertation Structure

The rest of the dissertation is broken into several chapters: background, related work, topological localization, geometric localization, and conclusion.

The background chapter discusses the different aspect of localization systems as well as some of the implementation aspects of this system. The related works chapter discussed other work on localization systems.

The topological and geometric localization chapters provide the approach and results of each type of system used for this work.

Lastly, the conclusion chapter discusses the contribution, limitations, and the future work.

Chapter 2

Background

The taxonomy of different methods of localization systems are discussed in this chapter as well as the resources used.

2.1 Taxonomy

Many types of localization systems have been developed. This section describes a taxonomy that encompasses most of these systems.

2.1.1 Sensors

The different sensors are used for localization systems are described in this section.

Vision Only Few systems use only a camera as their only method of sensing the world. Two such examples are [68] and [69]. Both of these approaches use a feature-based method where the system is trained to recognize features and place the system at a specific location. Features are parts of an image that can be used to describe the image such as corners, edges, and color. Feature descriptors are

a vector of features that are used to describe the entire image so that it can be distinguished from other images.

There are different methods of using cameras, including using a panoramic camera to capture more environment features with a single image such as [69]. Stereo vision has been used in [52], and [27] uses a single camera to detect artificial landmarks in an indoor environment.

A brief comparison of some vision-based methods can be found in [57].

Other Sensors Most localization systems use sonar [36] [50], laser range finders [49] [37] [43] [39], or cameras [30] [27] [69] [36] [49] [70] [43] as their method of sensing the world. These methods usually add odometry and other sensors in order to increase the accuracy of the system. Odometry has been determined to be inaccurate over long periods because of wheel slippage and the accumulation of error over time. Using probabilistic methods, odometry error can be dealt with effectively and the overall accuracy of the system can be improved.

Many systems use a combination of sensors including range finders, sonar, cameras, and Global Positioning Systems. [49] describes that using a single sensor can cause problems and how multiple sensors allow for a more robust system, and the integration of vision and range to create a robust indoor localization system. The system described in [43] uses a combination of all the stated sensors except for sonar. That system also includes odometry, a compass, and a wireless local area network (LAN) to communicate with other robots which all help to localize the robot.

2.1.2 Global Localization and Position Tracking

Position tracking, also known as local and relative localization, refers to the problem where the initial position is known and the position must be tracked relative to the initial position. These usually rely on compensating for odometry error. Kalman filtering is a popular method for solving the position tracking problem.

Global localization, also known as absolute localization, refers to the problem where the initial location is not known. Systems that solve this problem are more robust and can recover after becoming lost unlike those that solve only the position tracking problem. There are several methods for solving this type of problem such as Markov localization [30] [16] [17], Monte Carlo localization [66], and Kalman filters [45].

2.1.3 Probabilistic and Non-Probabilistic Methods

Most probabilistic methods use Bayes Rule as their basis for computing probabilities. Probabilistic methods using Bayes Rule are described in [18] [56] [63] and [23]. The main goal of the probabilistic systems is to compute the posterior $P(location = l \mid sensordata = s)$. This is the probability of being at location l given sensor data s. The systems start out with a probability distribution $P(location = l)$ and modifies it based on evidence and actions of the systems. To simplify, this will be denoted as $P(l)$. Bayes rule calculates $P(l \mid s)$ as $P(s \mid l)P(l)/P(s)$. The rule is typically simplified to:

$$P(l \mid s) = \alpha P(s \mid l)P(l)$$

where α is a normalizing factor to normalize the total probability to one.

Probabilistic Methods

Some of the more common probabilistic methods that are being used in most localization systems include Kalman Filter, Markov, and Monte Carlo.

Kalman Filter The kalman filter treats all probabilities as a Gaussian distribution [23], i.e. $P(l)$ and $P(s \mid l)$. The kalman filter is a position tracking algorithm, so the initial position must be known, and it compares favorably in terms of efficiency to other probabilistic methods, but it cannot reliably solve the kidnapped-robot problem nor global localization [23], described in later sections.

Markov Markov localization is a probabilistic method in which the environment is typically assumed to be static. The approach described in [17] uses filtering to improve results in a dynamic environment.

Another assumption made by the Markov approach, called the *Markov assumption* [55], specifies that if one knows the robot's location, then future measurements are independent of past measurements.

$$P(X_{t+1} \mid X_{0:t}) = P(X_{t+1} \mid X_t)$$

The next state is dependent only on the current state, i.e. a first-order Markov process.

Markov localization maintains a position probability density over the set of possible poses [17]. In cases where the location is unknown, the density may be uniformly distributed over the set of all possible poses. If the location is fairly

certain, then the distribution is centered around the known location. This allows the algorithm to globally localize and to deal with the kidnapped robot problem.

A more detailed description of the Markov algorithm can be found in [17]. Some works that use the Markov algorithm are described in [30] [8] [58] and [50]. [9] describes a variation of the Markov algorithm that dynamically changes the search space, acting as both a position tracking algorithm and a global localization algorithm.

Monte Carlo The Monte Carlo algorithm, also known as *particle filters* and *condensation algorithm*, represents the belief by a set of samples or particles [66]. The set represents the posterior of the belief. The samples are chosen randomly, but from an area that represents the desired distribution. For example, a distribution that represents the location of being one meter from a known landmark has samples randomly chosen that center around the landmark, with a radius of about one meter.

The samples are weighted in terms of probability; and the motion model is applied to each sample after an action until the set is decreased to a distribution that circles the most likely position. A more detailed description can be found in [12] and [66].

Non-Probabilistic Methods

Many non-probabilistic methods exist for localization. The system described in [25] uses a laser beacon placed on walls and a photo detector to determine the location and orientation based on the time and angle of the beam emitted. [69] uses a nearest-neighbor algorithm which compares the locations of only the neighbors of the previously believed location to the current sensor data. The map

is an adjacency graph (topological map, described later) where the neighbors are determined by adjacent nodes in the graph.

2.1.4 Active and Passive Localization

Passive localization systems do not have control over the motion of the robot, they simply perform localization based on the incoming sensor data [16]. Active localization, however, assumes that the localization routine has some control over the motion of the robot, providing an opportunity to increase the efficiency and robustness of localization [16]. Active localization systems, in other words, can move the robot to specific locations with the only purpose of performing localization.

Most systems use passive localization. Most navigation systems assume that the location of the robot is known and plan their path from there. Two active localization systems are described in [16] and [52].

2.1.5 Geometric and Topological Maps

Localization methods can be classified by the type of map they use. The current methods use geometric maps, topological maps, or a hybrid approach. Geometric maps are usually two-dimensional grids similar to that of a floor plan, topological maps are adjacency graphs, and hybrid approaches use both methods [69].

Geometric Maps

The exact position can be tracked using geometric maps (also known as metric and grid maps) with respect to the map's coordinate system [69]. These maps are usually used with landmark detection or map matching algorithms.

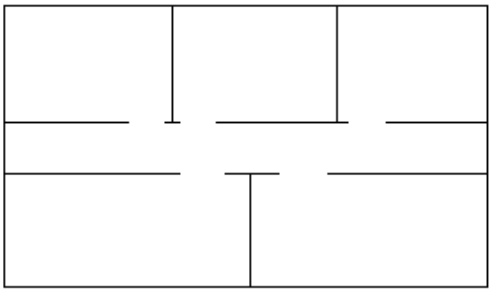

Figure 2.1: A geometric map using lines to represent walls

Feature maps and occupancy grid maps are examples of geometric maps. Feature maps use lines, walls, and corners as some of the features in the map. An example of a feature map is shown in Figure 2.1. Occupancy grids divide the environment into cells, and each cell contains information as to whether it is occupied or not [28].

Landmark detection systems use either artificial or natural landmarks. Artificial landmarks require changes to the environment but are usually easier to detect. Natural landmarks are better for environments where changing the environment is difficult, not allowed, or the environment is very large.

Artificial Landmarks Artificial landmarks are not often employed because they require a change to the environment. They are easier to locate with more accuracy, but the changing of the environment is not usually the preferred method, and artificial landmarks may not be feasible in very large environments. However, some environments like a hospital where lines may be painted on the floor or ceiling lend themselves better to this type of method. [27] is an example of a localization system that uses artificial landmarks in an office building.

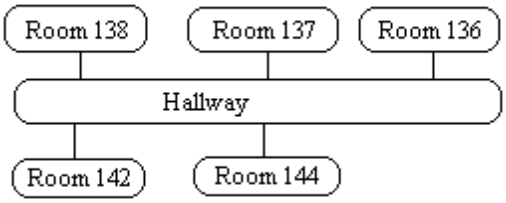

Figure 2.2: An example of a topological map

Natural Landmarks Natural landmarks are the most widely used because of the limitations of artificial landmarks. Several systems that use them are [36] [6] and [32]. Indoor natural landmarks may be something as commonplace as doors, windows, or corners. An outdoor landmark may be something such as a building, a large tree, sidewalks, and roads.

Map Matching This method tries to determine location by matching sensor values to the given map. Some examples that use map matching are [37] and [39].

Topological Maps

The topological map (graph), shown in Figure 2.2, is made up of nodes where the edges correspond to possible routes to other locations. The goal with these maps is to determine which node the robot is in. The advantage of this kind of approach is that the maps can be built more easily, but the location that the robot is localized is more coarse than the geometric maps which give a precise location. Some topological approaches are described in [69] and [30].

Hybrid

Most localization systems are now built using a hybrid approach. As described in [67], using one or the other can cause a great deal of problems for a localization system. Topological maps allow for simpler navigation, but often do not have enough information to localize to a fine area like geometric maps. Geometric maps have difficulty scaling to large environments. Some hybrid approaches are described in [67], [36], and [39]. A hybrid map includes both geometric and topological information. The metric information can include size of the node, direction and distance to neighboring nodes, or an entire geometric map such as an occupancy grid.

[67] describes a method of building topological maps from a geometric map. This allows for creating the geometric map normally, but still being able to use topological methods without manually creating the nodes.

2.1.6 Location

Localization systems are usually built for a specific location. The localization systems usually require a map of some type which gives a description of the world. Most localization systems work in a well structured environment such as indoors; others rely on GPS to localize and therefore only work outdoors. Some robots build a map of their environment as they move and use the same map to localize.

Indoor Only Most localization methods fall into the indoor only category, such as [50], [30], [49], [70], and [49]. The structured environment allows for simplifying assumptions such as always being on a flat surface. Many of these methods

use straight lines from features such as corners, windows, desks, doors, etc to determine their location and orientation.

Outdoor Only The localization methods typically used only for outdoor robots usually include GPS as the main method of localizing, such as [43], [47], [10], and [62]. However, since GPS does not always receive enough satellite signals when in areas that block the signal or the GPS does not give accurate results, odometry is also used in order to increase the accuracy and to supplement the GPS in case of GPS failure.

Indoor and Outdoor Few methods use both indoor and outdoor environments. This is because the indoor methods which assume a highly structured environment do not work outdoors and GPS does not work indoors. Finding a method that works equally well outdoors and indoors can be difficult. A few methods which report to work well in both are [69], [37], [39], and [29]. Two of these, [69] and [29], use an appearance-based method of localization which uses camera images to capture the layout and texture (appearance) of the area rather than specific landmarks in the environment.

2.1.7 Mapping

There are two different methods of creating the map used for localization. The most common is to create a map before trying to perform localization. The other method is to have the robot create the map as it moves through the environment, also known as SLAM.

SLAM The SLAM (simultaneous localization and mapping) problem is one that has been worked on more recently in the robotics community [44] and [47]. These methods allow a robot to operate in an environment for which there is no map, i.e., an unknown location. This method can be used for some service robots such as wheel chair robots, floor cleaning robots, and search and rescue robots.

There are several issues that must be addressed when performing robotic mapping: the correspondence problem, dynamic environments, and robotic exploration. The correspondence problem is the problem of determining if sensor measurements taken at different points in time correspond to the same physical object in the world [64]. Dynamic environments include objects that may move such as cars and people, and natural changes which occur with changing seasons. Robotic exploration is the problem where the robot must choose its path while mapping the environment [64].

SLAM methods can still make assumptions about their environment. For instance, the floor cleaning robot will assume that it is inside when creating its map, and therefore will use features that are specific to an indoor environment.

The advantages of this methodology is that a robot can be placed in an environment that has not been visited before by the robot or humans in general, and that the map of the environment can be continuously updated so that changes that occur over long periods will not affect the localization as much.

The disadvantage is that some robots with a specific purpose such as a delivery robot or a tour guide must know where specific locations are before they are placed in the environment in order to find a specific location, for instance building 4, room 312. The exploring of the environment will probably not yield enough information in order to classify specific locations. If exploration is used, then specific places would have to be manually labeled. Also, the exploration stage

must have limits on where the robot can go, making this method more suitable for indoor environments.

Non-SLAM Systems that do not use a SLAM methodology must have a map prior to operating. A number of works discussed so far such as [39], [10], [25], and [49] do not use a SLAM methodology.

Creating the map can take a significant amount of work, and the map may contain errors. Some systems are built with the assumption that the map may not be accurate and usually account for this using probabilistic methods.

Maps for large environments built using manual methods can take a long time to generate. Static maps in dynamic environments can also pose a problem. One method for dealing with this might be to have a partial-SLAM method where the robot is given a map of the environment with minimal information, but the map is updated by the robot as it moves through the environment.

2.1.8 Scalability and Generalizability

Many localization systems can work well when moved from one environment to another, e.g., from one office building floor to another, as long as the map is supplied, and the type of environment does not change, for example, from indoors to outdoors.

Many indoor systems such as [49] rely on specific indoor features such as straight lines from walls, desks, and windows. Many outdoor systems such as [47] rely on GPS for localization. Sensors such as laser range finders and sonar require an environment without wide open spaces. The system described in [36] uses sonar and odometry and requires "corridor-rich" environments like those found in hospitals and office buildings; however, the system is able to operate indoors as

well as outdoors.

The scalability of these systems varies. "Large-Scale environments" are defined in [34] as "if its spatial structure is at a significantly larger scale than the sensory horizon of the observer." Many systems evaluate their system over in an area larger than one floor in an office building. [36] reports their system operates over a large-scale area where testing is performed over an entire university campus.

2.2 Resources

This section discusses the resources used to develop the proposed localization system.

Tools The OpenCV computer vision library [51] was utilized for several aspects of the system, such as principle component analysis, opening and writing images, and many other functions. The library is an open source computer vision library that is widely used.

The Cluster program [7] was also used as part of this work. The program was used for the estimation of the Gaussian Mixture Models (GMMs) used in this work.

David Lowe provides a program that generates SIFT keypoints from PGM images. This program was downloaded from his webpage and used for SIFT keypoint generation [42].

Computing Platform The localization system was written using C++ on a machine with four dual core Opteron processors and 20 gigabytes of RAM. The localization software was not written to be multi-threaded however. The GNU C++ compiler was used.

Robots The P3-AT robot platform from ActivMedia [54] was used to capture images. It was operated under remote control and included a Sony EVI-D30 PTZ camera. Images from the camera were captured at a resolution of 640-240 15 frames per second. This was the largest resolution the capture device would allow.

Chapter 3

Related Work

The related work is described in this chapter to provide a brief background of some of the projects that have performed research in the area of localization and mapping.

3.1 Mobile Robotics

There are several robots that use localization as a key part of their navigation. DERVISH was designed by researches at Stanford University and won the Office Delivery event of the 1994 Robot Competition and Exhibition [50].

DERVISH was an indoor operating robot and used sonar as its main method of sensing the world. The sonar was placed so that it could detect both short objects and tall objects like a shelf that it might not fit under. For the competition, each robot was given a topological map of the office and a goal room that had two different doors to enter.

A Markov probabilistic algorithm was used to determine its location based on features that DERVISH would detect, such as open and closed doors, hallways,

foyers, and walls. A probability table was given for each of the features that gave the likelihood of detecting each feature when that feature appeared, as well as the likelihood of detecting it as another feature. For instance, the probability of detecting an open door as an open door was 0.9, and the probability of detecting it as a wall was 0.05. With five features, the table had 25 probabilities.

The robot did not use odometry, it used events to determine when to update its state. An event happened when the sonar detected one of the features listed. When an open door was detected, the Markov algorithm updated every possible state. Every state must be updated because the sonar might have missed detecting some features. When a feature was detected, the robot was possibly in a new node. Without odometry, it was possible to move a long ways and miss several features.

Researchers at the University of Bonn, Aachen University of Technology, and Carnegie Mellon University worked together to design Rhino which was deployed at the Deutches Museum in Bonn, Germany [8]. The robot used four sensor systems: laser, sonar, infrared, and tactile. It relied on the laser range finder for localization. The software consisted of 25 modules which ran on three on-board PCs and three SUN workstations which were off-board.

The localization system used a metric map and the Markov algorithm. Because the robot was deployed in a museum, the people surrounding the robot made the environment very dynamic. This violated the Markov assumption of a static environment. Therefore, filters were used to sort the measurements into corrupt and uncorrupted categories It did this by determining if the measurement increased or decreased the certainty of the robot. Measurements that did not increase the certainty were assumed to be corrupt and were not used to update the belief.

An occupancy grid map was used as the metric map. The map approximated

the probability that a grid on a discretized approximation of the environment was occupied. The map was discretized into 2D grids which determined how fine grain of accuracy was needed.

Afterwards, a variation of the Markov algorithm, called the Dynamic Markov Localization (DML) algorithm [9] was implemented for the robot. This algorithm differed in that it attempted to perform both position tracking and global localization. It performed position tracking by reducing the amount of state space that the algorithm had to search over. In situations where the robot was almost certain about its position, i.e., the distribution was centered around one location, the remaining states had extremely small probabilities. DML used an octree to represent the state space, where states with extremely small probabilities (less than a threshold) were grouped together. An octree is a structure that spatially divides a three-dimensional space into cubes of varying size into a tree-like structure. The states in this grouping were updated only once, applying the same update to every one.

The algorithm also simultaneously calculated the likelihood that the robot's position was not contained in the currently considered states. If this happened, more states could be considered by changing the octree. This allowed for a dynamically evolving state space that was considered based on the certainty of the robot, thus improving efficiency when the robot was certain about its location.

Minerva, created after Rhino, is the second version of a museum tour guide robot [65]. It was deployed in the Smithsonian and required some improvements in order to successfully operate in the significantly larger museum with significantly more people.

Localization still relied on a laser range finder as the main sensor, but a camera which pointed at the ceiling was used to augment the system because of the wide

open spaces. A texture map of the ceiling was created and this was used to localize using the camera.

Minerva also had the ability to learn its maps from scratch whereas Rhino was given a manually created map with no ability to create its own.

Researchers at the University of Bremen developed the service robot Rolland as part of the *Bremen Autonomous Wheelchair* [35] project. The localization system, described in [36] works over large-scale environments and was tested on the campus of the University of Bremen.

Rolland is different from the other robots discussed so far in that it can move both indoors and outdoors. However, the robot is limited to corridor-like environments as described in [36]. The robot does not perform any mapping, so an a priori map is required; however, it is stated that this will be done in the future.

The robot uses topological maps which are called route graphs, where the nodes correspond to decision points such as hallways, corners, and junctions. The edges of the topological map represent straight corridors that connect the decision points. The graph integrates geometric information such as the length of the edges (corridors) and the angles of decision points and corridors.

The localization method uses odometry and sonar to detect corners. The approach is to match the current traveled route with the route graph in a probabilistic manner. The robot currently does not perform global localization, instead it must be in a known initial position to work. It must also have a known route so that it can direct the user to the goal location.

The service robot PSR1 developed at the Korean Institute of Science and Technology uses the Monte Carlo algorithm, and the probability is computed using a map matching algorithm which compares the laser range finder scan with

those stored in the map [37]. The comparison algorithm uses two functions to compare the results, a range image similarity function and an angular similarity measure function.

The robot proved to work well in somewhat dynamic indoor environments using the Monte Carlo method and map matching algorithms.

Map matching is also used in the Kurt3D robot, developed in Germany at the Fraunhofer Institute for Autonomous Intelligent Systems [39]. The algorithm is designed for speed, allowing localization of the robot moving at four meters per second. The robot uses both stereo vision and a laser range finder; and the localization routine works both indoors and outdoors.

The robot uses a position tracking method, which means that it cannot recover after getting lost. Because position tracking algorithms are typically more efficient than global localization algorithms, this was necessary for the high speeds at which the robot moved.

An indoor localization system based on a laser beacon (emitter) and receptor has been built by researchers at the University of La Laguna [25]. Beacons are placed on walls in hallways and rooms that sweep rooms with the laser and a receptor on a robot receives the beams. The pose within the room is determined by the time and angle to the robot.

The beacons are somewhat inexpensive but require that every room in the building to be modified. The beacons must be placed so that objects such as tables do not block the beam. Some rooms may require multiple beacons. The system is very fast and accurate, but may not be suitable for large buildings. The system is not suitable for outdoor environments. Also, the robot may get lost if just one beacon fails to work, so the system is not very robust in being able to recover from failure.

A distributed outdoor localization system was built at the Oak Ridge National Laboratory that uses multiple robots to cooperate in order to achieve a goal [43]. The ATRV mini robots use several sensors including differential GPS, a pan-tilt-zoom camera, a compass, a laser rangefinder, a wireless LAN, and odometry. A Kalman filter is used as the position tracking localization algorithm. When one of the robots can no longer localize because of deteriorating sensor data, another robot can help it localize.

This system does not work indoors. Also, the sensor suite of this system makes the robots very expensive. However, having several robots able to help others localize makes this system very robust to failure of the sensors, especially GPS which can lose signal with satellites because of buildings, trees, and many other reasons.

A similar robot [10] also uses a Kalman filter along with differential GPS and odometry in order to localize outside. This robot is designed for volcano exploration but does not use a distributed localization system.

GPS can be a very accurate sensor, however, many environments are not suitable for relying on GPS alone because of the possibility of losing satellite signals. Using GPS automatically gives access to a hybrid like map. This is because coordinates can be viewed as nodes of a map with geometric information built in as the distance and direction between nodes can be determined. Many drivers of trucks and cars already use GPS to travel great distances. Differential GPS is not required in these applications as the accuracy of normal GPS is typically good enough to determine what road the driver is on.

Researchers at the University of Munich state that using a single sensor is not robust enough for localization [49]. The system they propose uses range and vision in order to make a more robust indoor system. The camera is used to find

vertical lines from windows, doors, corners, etc. and the range finder determines the location of walls in front of robot. A Kalman filter is used to determine the location; then the image and range are used to determine a more precise location. This system is another position tracking localization system that has the same problem of not being able to recover from being lost, but the system is reputed to be accurate in finding the location within 10 cm and orientation within 15 degrees.

3.2 Appearance-Based Localization

At the University of Amsterdam an active appearance-based method has been used for localization of a robot named Lino [52]. Active systems have some control over the navigation of the robot in order to move to a location to make the system more certain about its location. For example, moving closer to a landmark in order to be certain about which object it is, and therefore being more certain about its position would be an active system.

The system uses a Monte Carlo probabilistic algorithm with stereo cameras on a pan-tilt device as the main sensors. Because appearance-based techniques can have trouble localizing in dynamic environments, their approach is to move the stereo head to look at a location that has not changed as much. Their appearance-based technique is based on using disparity maps. The disparity map is a two-dimensional depth map similar to that of a laser rangefinder one-dimensional map, but with one more dimension. Features can be selected from the disparity map and compared with the disparity maps that are stored in the map of the environment, similar to a map-matching technique.

They found that edges extracted as features work well for dealing with illumination changes in the environment. Using this approach, along with the

active vision technique of looking for a less changed location makes for a robust algorithm that works well. Because the system is dependent on depth maps, this system may not work as well in all outdoor areas. Outdoor areas that have lots of structures should work, however.

An appearance-based method that is insensitive to illumination changes is proposed in [29]. The researchers use an omni-directional camera in order to view more of the environment at a time. This gives more features to compare with a map at any given time.

Eigenimages [38] are used to define the environment in this method. Eigenimages represent a set of training images in an eigenspace. If the images are highly correlated, the dimension can be reduced significantly using principle component analysis. Images that are not part of the training set can be projected onto the eigenspace. The coefficients from this projection are compared with those of the training set by determining the smallest angle between them (using a dot product).

Researchers at Carnegie Mellon University have developed an appearance-based approach that also uses an omni-directional or panoramic camera [69]. A topological map is used and defines the locations of the images. Training images are taken by using a camera to retrieve images while going through the environment, and grouped later into their locations. Color histograms are created from the images from two color spaces, RGB (red, green, blue) and HLS (hue, luminance, saturation). Color histograms are vectors that count the number of specific pixel colors that appear in the image: one vector for each band of color. A nearest-neighbor approach is used for comparing images. The color histogram created from the current image in the environment is compared to only the location where the robot is believed to be in, and its neighbors. A voting mechanism is used to determine to which location the image belongs.

This algorithm is similar to a non-probabilistic position tracking algorithm in that it cannot recover from getting lost and it cannot perform global localization, as it must know its initial location. The algorithm was tested over a small area that included both outdoor and indoor locations, and it was reported to have performed reasonably well.

Chapter 4

Topological Localization

The first step to solve the localization problem is topological localization. This chapter describes the methods and the results of attempting to localize the system using the topological map.

4.1 Approach

The approach proposed in this work is to use only a camera to perform localization in multiple environments including indoors, outdoors, and at polar camp sites in Greenland and Antarctica. Therefore, a topological localization system was designed that uses appearance-based features and a hidden Markov model as the classifier [2].

The topological map was created manually of an area inside and outside a single building. Each node of the map is a set of images that are representative of the location. After the images for the map are acquired, the appearance-based features are extracted and modeled using a Gaussian mixture model. After this step, the topological map is represented by a Gaussian mixture model. The

likelihood that an image is part of a specific node in the map is generated by extracting the proper feature from the image and applying it to the Gaussian mixture model. All the processing in this work was done offline to simplify the testing.

4.1.1 Appearance Based Method

The appearance-based method uses the appearance or texture of an image in order to recognize a location. Appearance based methods have been used in [69], [52] and [31]. Instead of using a geometric (explicit) representation where the images are used to find walls or objects, the images themselves represent the model of the environment.

Some of the feature descriptors that were used include color and gray histograms [22], Hu Moments [22], and eigenimages [38] and [48].

Pixel values in images usually range from 0-255. A gray histogram is a count of all of the pixel values in the image. The color histogram is the same, except the count exists for all three bands of the RGB images. The histogram in this case is actually three separate histograms.

Images can also be described using statistical moments such as mean, variance, skewness, and higher order moments. [26] describes a set of seven moments that are invariant to rotation, translation, and scale changes. These moments are referred to as Hu Moments.

Eigenimages are a set of basis images that are used reduce the dimensionality of an image. The eigenimages are created from a set of images. Images are projected onto the eigenimages to give a descriptor that is much smaller than the image itself.

Gaussian Mixture Model

A Gaussian mixture model (GMM) was used to model the feature descriptors. A Gaussian mixture consists of a linear combination of Gaussians (normal distributions). Each Gaussian in the mixture has its own weight and the final probability is given by linearly combining the results of each Gaussian in the mixture. The EM [13] [55] algorithm is used to generate the parameters of the mixture which include the mean, covariance, and weight. The program described in [7] was used to generate the Gaussian mixture models for this work and also gives a good description of the Gaussian mixture model. The GMM is used to generate the likelihood probabilities in this testing, as described in Equation 4.1.

$$P(X = x|Q = q) = \sum_{k} w_k * N_k \tag{4.1}$$

Where x is the current feature, q is the current location, k is the number of Gaussians in the mixture, w is the weight of the k^{th} Gaussian, and N_k is the k^{th} Gaussian in the mixture. This is the probability of seeing the feature x at location q.

A GMM is typically used to model data where the normal distribution does not work well by itself. The GMM can work well even in cases where the data is not normal, or the assumption of normal data is incorrect.

4.1.2 Hidden Markov Model

The hidden Markov model (HMM) is one of the simplest bayes networks. It consists of a set of N states, the initial probability distribution, and a set of transition probabilities. The HMM can be used to model temporal data. It has been used extensively for speech recognition [53]. The HMM relies on the Markov

Assumption, that the value of the next state is dependent only on the values of the current state [55].

The hidden Markov model decoding algorithm described in [14] on page 135 was used for determining which node in the map has the highest probability. This is the problem of determining the sequence of hidden states given a sequence of visible states. The visible states in this case are images captured from a camera and the hidden states are the locations in the map.

The results for the localization are compared with those of two other simple classifiers, ML (maximum likelihood) and a variation of the HMM where all the transition probabilities are equal, thus *turning off* the hidden Markov model and acting more like the naive Bayes classifier. The ML classifier selects the node with the highest likelihood (from the GMM) and has no memory, see Equation 4.2. The naive Bayes algorithm stores a posterior probability. The likelihood for a node is multiplied by its posterior, and then the node with the highest posterior value, maximum a posterior (MAP), is chosen as the selected node, see Equation 4.3.

$$\text{ML Location } = argmax[P(X = x|Q = q)] \tag{4.2}$$

$$\text{Naive Bayes Location } = argmax[\alpha P(X = x|Q = q) * P(Q)] \tag{4.3}$$

Transition Probability

The transition probability was modeled using a zero mean normal distribution based on the distances of nodes in the topological map. The variance was a preset constant parameter, 1.0 in this case. A minimum value was set so that once the

Distance	Weight
0	1.000
1	0.607
2	0.135
3	0.011
≥ 4	0.001

Table 4.1: Transition weights for the Hidden Markov Model. All weights are based on the shortest distance from one node to another. The weight is the total probability given to all nodes at a given distance. These values are the same values used for this work. These are based on a zero mean Gaussian with a variance of 1.0 and minimum probability of 0.001.

probability went below that minimum value, all distances from that point were given the same minimum value. These values represent the probability of moving from one node to another node, with the highest probability being to stay in the current location; and the probabilities progressively getting lower the further away the node is from the current node.

The probability given for a specific distance was used as a total weight value for all nodes of that distance to add up to. For example, if there are four adjacent nodes j to node i, and the probability of being at distance one is 0.80, then the probability of moving from node i to node j is 0.20. After determining the probabilities for the transition from node i to every node in the map, the values are normalized to 1.0.

Using the probabilities as a total weight is important because it is possible to have the probability of moving to another location be higher than staying in the current location. This can happen if the current location has many connections. This would mean that each of these connections are at distance 1.0. If the probability of moving to a location of distance 1.0 is 0.8, then the total probability of moving to an adjacent location can be higher than not moving, which is not desirable. Therefore, normalizing the total probability of all the adjacent locations

to 0.8 helps to prevent this from occurring. The actual distance weights used in this work are included in Table 4.1

The distance is determined by the fewest number of connections from one node to another. This algorithm requires finding the shortest distance to other nodes on the map. Moreover, depending on the variance and the minimum probability selected, the distances only had to be calculated for the nodes up to a small distance (three in this case).

4.1.3 Topological Maps

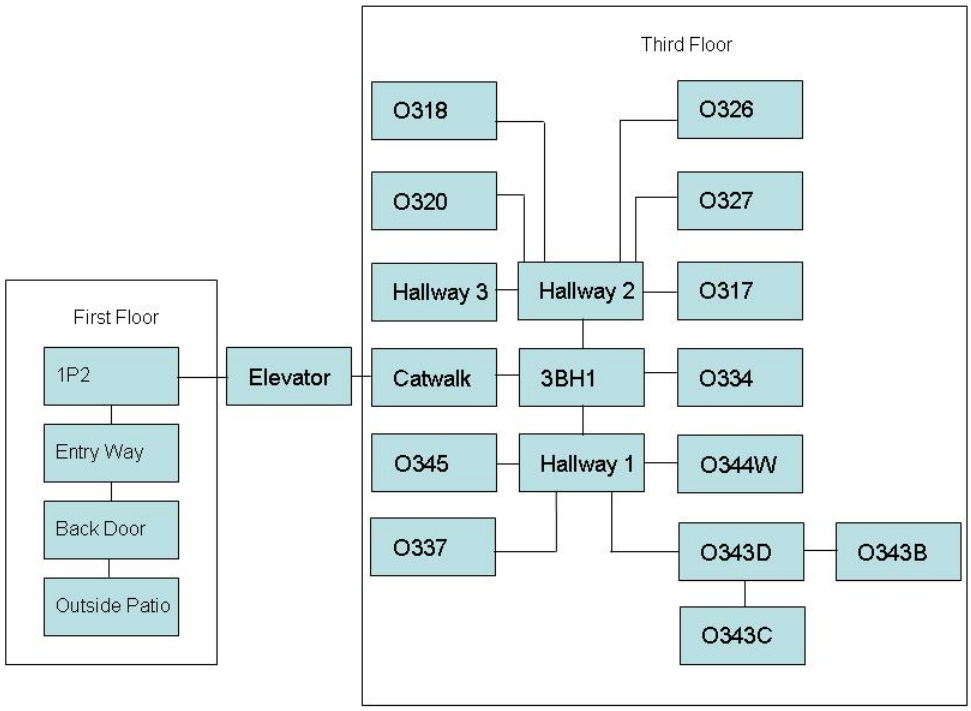

Figure 4.1: Adjacency graph representing the first topological map used for evaluation. The building has three floors, but the map was created from locations on only the first and third floors.

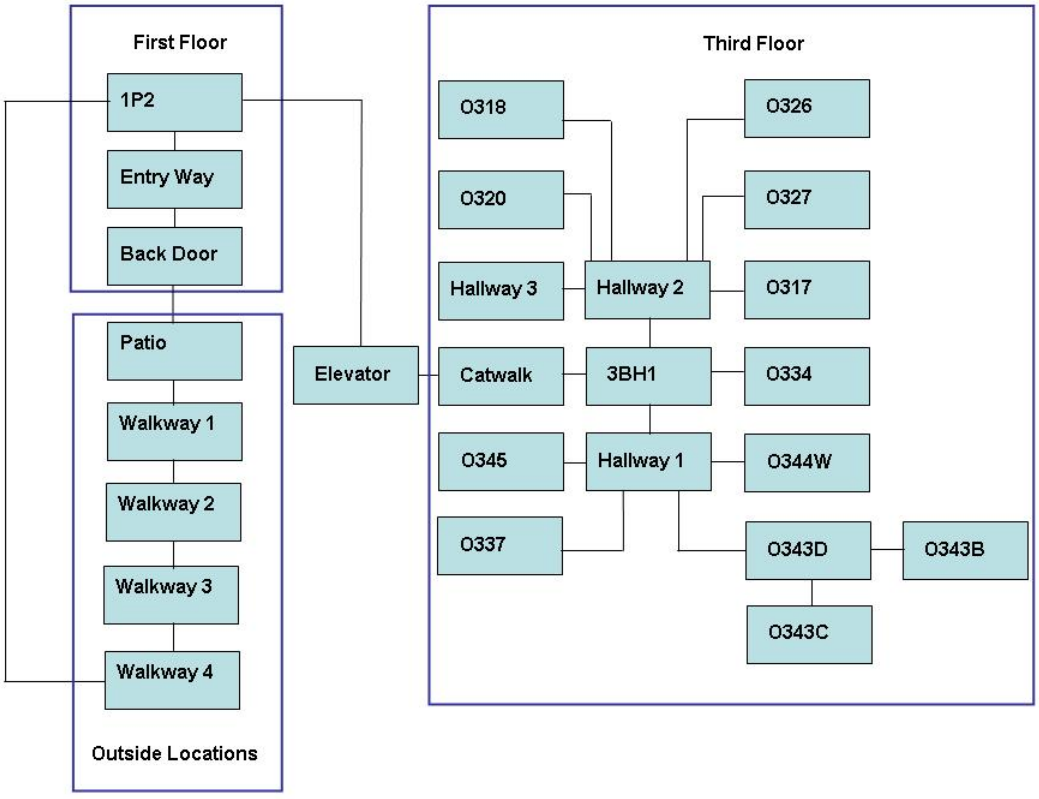

Figure 4.2: Adjacency graph representing the second topological map used for evaluation. This map is broken into three different general locations: first floor, third floor, and the outside areas.

Two different maps were used to test the system. The first map, see Figure 4.1, was a smaller version of the second, not including many of the outdoor locations. The locations to use in the map are from one building. They were chosen based on availability and accessibility. Once a location was chosen, a robot platform with a camera was driven using remote control through the areas while capturing and saving all images. These images were grouped together by location and became the database for all tests performed. The first map has 22 locations made up of over 22,000 images total. The map contains 12 offices, 10 hallway locations, and 1 elevator that span over two floors of the three floor building. Figures 4.3, 4.4, 4.5, and 4.6 show a sample of some of the images used to make up inside nodes 334, 335, Hallway 1, and Hallway 2, respectively. Figure 4.7 shows images from the outside Patio location. Each image was captured at a resolution of 640x240.

The second map has four more nodes than the first. The map was built using over 26,000 images. These added nodes are all outside locations, shown in Figure 4.2 as Walkway 1, Walkway 2, Walkway 3, and Walkway 4. These nodes are connected from the patio back to the front entryway, 1P2. These were added because the single outside location from the first map, Figure 4.1, does not give a good indication of how well the system works outdoors. Figures 4.8 - 4.11 show images from the new outside locations.

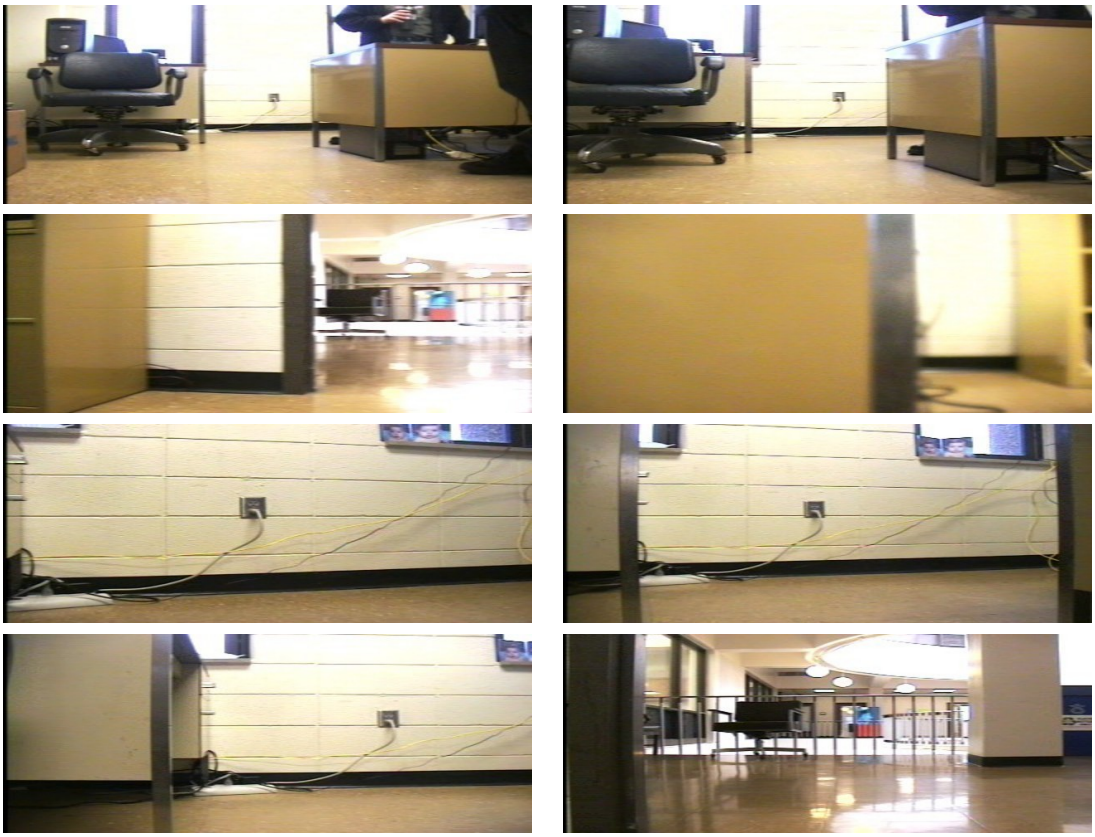

Figure 4.3: Room 334 sample images

Figure 4.4: Room 335 sample images

Figure 4.5: Hallway 1 sample images

Figure 4.6: Hallway 2 sample images

Figure 4.7: Patio sample images

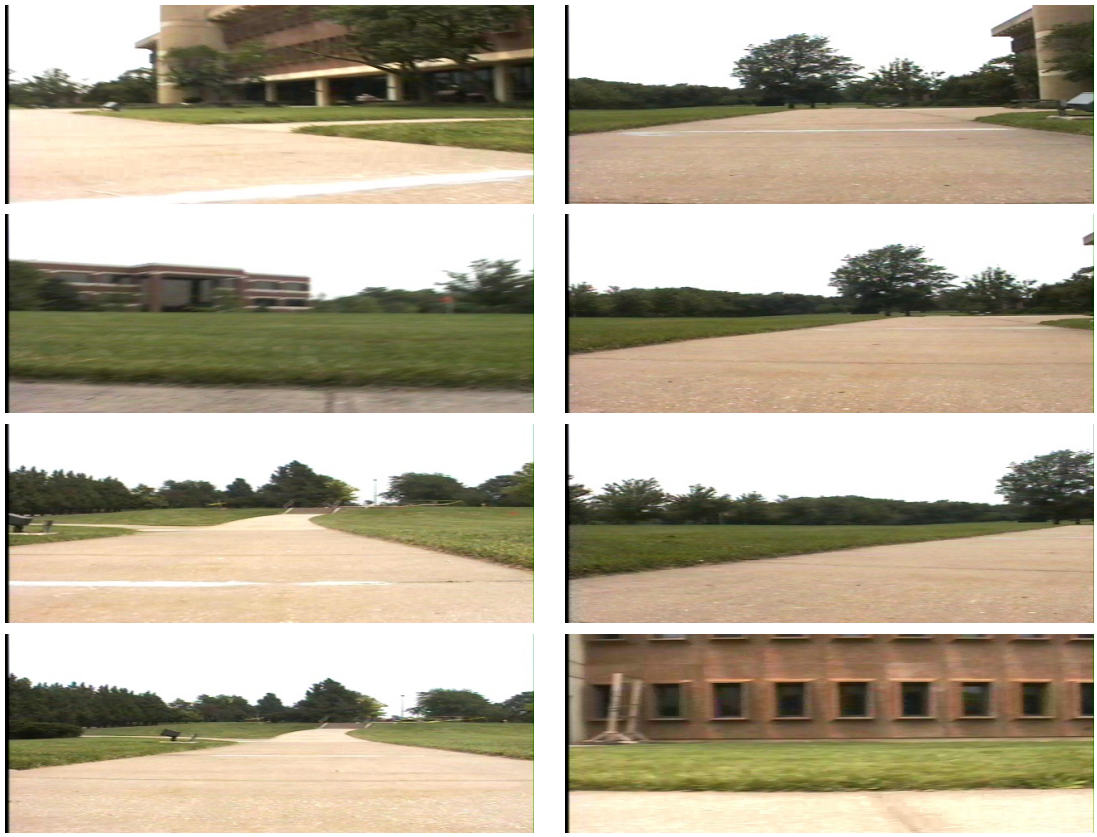

Figure 4.8: Walkway 1 sample images

Figure 4.9: Walkway 2 sample images

Figure 4.10: Walkway 3 sample images

Figure 4.11: Walkway 4 sample images

4.1.4 Features

Several feature descriptors were used to test the system, namely, color and gray histograms, Hu Moments [22], and eigenimages [48] and [38]. Variants of the color and gray histograms were also used. Table 4.3 includes a list of all the descriptors used for this work.

In order to get a single descriptor for the color histograms from the RGB images, a histogram for each band of the RGB image is obtained separately then appended to the end of the previous one. Therefore, each color histogram is three times the size of its corresponding gray histogram. A variation that was used was to divide each image up into equally sized columns and rows. Then a separate histogram (color or gray) was calculated for each section of the image, again appending each descriptor to the end of the previous. This was done on sizes of 2x2 and 3x3.

Also, the number of bins used to calculate the histograms could be varied. The number of bins determines the size of the histogram. For example, 256 bins can be used for an image with pixel values ranging from 0 to 256. However, 128 bins can also be used causing a loss of information. This combines the pixel values of zero and one into a single bins in the histogram. This happens throughout the entire range of the histogram in this case. This is mainly used to reduce the amount the information in the histogram.

The histograms of the 1x1 and the 2x2 all used 256 bins, and the 3x3 descriptor was calculated on 128 bins in order to try and reduce the size of the descriptor. Table 4.2 lists all the variations of the histograms that were used for testing.

Two versions of the Hu Moments descriptor were used. One descriptor is calculated from a gray scale image, the other calcuated each of the seven Hu

Color/Gray	Rows	Cols	Bins	Size
Gray	1	1	256	256
Color	1	1	256	768
Gray	2	2	256	1024
Color	2	2	256	3072
Gray	3	3	128	1152
Color	3	3	128	3456

Table 4.2: Variations of the color and gray histogram feature descriptors

Moments on each band of the RGB image separately, and appended them similar to the color histograms. Therefore, the color Hu Moments descriptor was three times the size of the gray Hu Moments descriptor.

The eigenimage descriptor is described in [48]. This descriptor uses each image in the training set as a single vector and principle component analysis is performed to reduce the size of the descriptor. It was calculated on gray scale images.

Principle Component Analysis

Principle Component Analysis (PCA) was used to reduce the dimensionality of most of the descriptors. It works by computing a set of basis vectors (eigenvectors) and eigenvalues of a set of features [14]. The eigenvectors that represent the greatest amount of information as determined by the eigenvalues, can be used to reduce the dimensionality of the original descriptors. A vector can be projected onto the basis resulting in a reduced description of the vector.

The number of eigenvectors represents the size of the reduced descriptor. A single value for each eigenvector is obtained by applying the dot product on the descriptor and the eigenvector. This is done for each eigenvector, resulting in a vector the same size as the number of eigenvectors.

Most of the features used in this work were reduced using PCA. A suitable number of components had to be determined that would adequately represent

the features and still allow distinguishing between the features. It was found [57] that 20 components were sufficient to describe the eigenimage descriptors. The color and gray histograms were reduced to a similar number, i.e. 25. Because this number worked well, testing with a higher number was not performed. However, it is intuitive that the smallest number as possible that still allows for recognition should be used. Extremely high dimension Gaussian mixtures can cause problems for the Hidden Markov Model [68].

The numbers used in this work are not optimum numbers that provide the best solution. They were numbers that seemed to perform well for this approach. The numbers used can depend on several factors, including the size of the original descriptor and the number of images, or the size of the map. Table 4.3 lists the sizes of each feature descriptor used, before and after PCA. Three of the descriptors, the Hu Moments descriptors and the eigenimage descriptor were not changed in this step. The Hu Moments descriptors were small enough, that a reduction of information was not necessary. The eigenimage descriptor already uses PCA to reduce its dimensionality, therefore it was not necessary to reduce it again.

	Descriptor	**Original Size**	**Size after PCA**
1	Gray Histogram 1x1 with 256 bins	256	25
2	Color Histogram 1x1 with 256 bins	768	25
3	Gray Histogram 2x2 with 256 bins	1024	25
4	Color Histogram 2x2 with 256 bins	3072	25
5	Gray Histogram 3x3 with 128 bins	1152	25
6	Color Histogram 3x3 with 128 bins	3456	25
7	Hu Moments (Gray Image)	7	7
8	Hu Moments (RGB image)	21	21
9	Eigenimages	20	20

Table 4.3: Feature descriptor descriptions. The size of the descriptors before and after PCA has been performed.

Figure 4.12: Bedrock Cafe in North Grip sample images

4.1.5 Polar Environments

After several trips to both Greenland and Antarctica, PRISM/CReSIS has numerous pictures of the different polar locations. Several pictures were selected to be used as part of a map with six different locations. The number of images for each location ranged from around 30 to 100. The entire map included 383 images for training after removing the test images. Figures 4.12 and 4.13 shows some sample images from two of the locations used in the testing.

In previous testing however, the number of images per location needed to be around 200-300 before obtaining adequate results. The tests were performed with three descriptors, two of which were described previously in Table 4.3:

Figure 4.13: North Grip sample images

Eigenimages and Gray Histogram 1x1 with 256 bins. The other descriptor chosen was a Color Histogram 1x1 with only 64 bins, which contains only 25% of the information that the same descriptor with 256 bins contains. The reason for the fewer number of bins used is because there is not enough images in the entire map to perform PCA on the larger feature descriptors. The Color Histogram 1x1 with 256 bins requires at least 768 images to perform PCA, many more than the map contained. The Hu Moments feature descriptors were not used because they did not perform adequately in previous tests.

4.2 Evaluation

The tenfold testing procedure was used to evaluate the system. The image database was broken into ten random sets. Each set needed to contain images from every location in the map, therefore each location was broken into ten random sets. These sets were combined with the sets from other locations to create an entire set of images.

Each of the sets were selected to be used as a test set with the other nine being used as its training set. Ten separate training and test sets were used to test the system. Every test was then run ten times, using each different test set with its corresponding training data. The results from each of the ten runs were averaged together to get an overall result. All of the results given from this paper are the result of the ten fold experiments.

As was stated earlier, most of the processing was done offline. The extracting of features, performing PCA, and modeling the features with a Gaussian mixture model were all done offline. The images used for the tests were taken from the test set. A predetermined route was selected. Then a random number of images were

selected from the pool of images for the specific locations to represent the images that would be captured from the camera as if the robot were moving through the map. The features are then extracted from these images, their size is reduced using the eigenvectors from the PCA step on the training set, and these features are written to a file to be classified in the order they were selected, allowing the same features to be classified several times on different tests. These steps are performed once for every test set.

The results from three different classifiers are given: hidden Markov model, maximum likelihood, and an approach similar to the naive Bayes classifier, described previously in Section 4.1.2.

The results from seven different tests are given. The first three tests are used to determine how well the system solves the global localization problem through a normal traversal of the first topological map. Tests four and five are used to determine how well the system can handle the kidnapped robot problem. This problem is similar to the global localization problem where the robot is placed in some random location in the map and it must determine its location, except that the robot is transported to a new location in a completely different part of the map after already localizing itself. The robot must be able to 'unlearn' where it believes it is at, and then 'relocalize' itself to the new location. The last two tests are similar to the first three, but are performed on the second topological map and are designed to test moving outdoors.

4.2.1 Local Testing

The results for the Bayes like classifier are almost identical to those of the maximum likelihood results. This is because the likelihood values dominate the

Test	Nodes Visited (in order)
Test 1	O343C, O343D, O343B, O343D, HALLWAY 1, 3BH1, HALLWAY 2, HALLWAY 3, HALLWAY 2, O320
Test 2	O343C, O343D, O343B, O343D, HALLWAY 1, O337, HALLWAY 1, 3BH1, O334, 3BH1, CATWALK3, ELEVATOR, 1P2, ENTRY WAY, BACK DOOR, PATIO, BACK DOOR, ENTRYWAY
Test 3	HALLWAY 3, HALLWAY 2, O326, HALLWAY 2, O317, HALLWAY 2, O318, HALLWAY 2, O320, HALLWAY 2, O327, HALLWAY 2, 3BH1, HALLWAY 1, O337, HALLWAY 1, O344W, HALLWAY 1, O345, HALLWAY 1, O343D, O343C, O343D, O343B, O343D, HALLWAY 1, 3BH1, CATWALK3, ELEVATOR, 1P2, ENTRY WAY, BACK DOOR, PATIO
Test 4	O343C, O343D, O343B, O343D, HALLWAY 1, 1P2*, ENTRY, BACK DOOR, PATIO, BACK DOOR
Test 5	O327, HALLWAY 2, O317, HALLWAY 2, O327, O327, BACK DOOR*, PATIO, BACK DOOR, ENTRY, 1P2, ELEVATOR, CATWALK3, 3BH1, HALLWAY 2, O327
Test 6	HALLWAY 3, HALLWAY 2, O326, HALLWAY 2, O317, HALLWAY 2, O318, HALLWAY 2, O320, HALLWAY 2, O327, HALLWAY 2, 3BH1, HALLWAY 1, O337, HALLWAY 1, O344W, HALLWAY 1, O345, HALLWAY 1, O343D, O343C, O343D, O343B, O343D, HALLWAY 1, 3BH1, CATWALK3, ELEVATOR, 1P2, ENTRY WAY, BACK DOOR, PATIO, WALKWAY 1, WALKWAY 2, WALKWAY 3, WALKWAY 4
Test 7	PATIO, WALKWAY 1, WALKWAY 2, WALKWAY 3, WALKWAY 4, WALKWAY 3, WALKWAY 2, WALKWAY 1, PATIO

Table 4.4: Nodes visited for each test run in the order they were visited. Tests 4 and 5 were the kidnapped robot tests. The asterisk represents the node in which the robot was 'teleported' to another location not directly adjacent to its previous. Tests 6 and 7 were performed on the second map.

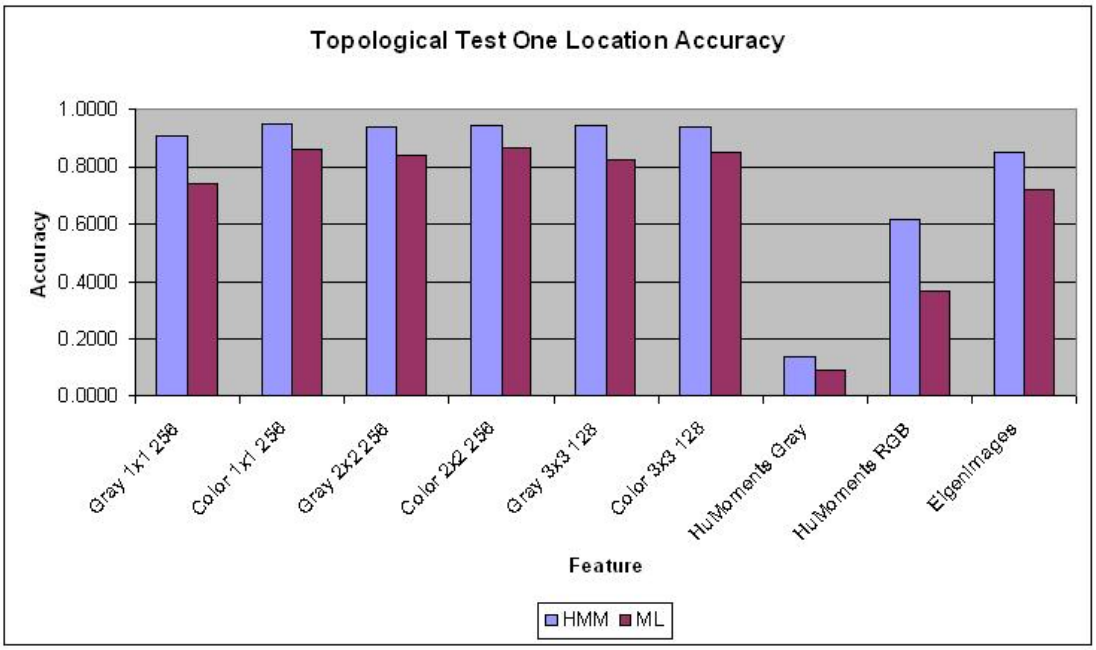

Figure 4.14: Topological test one: position accuracy

posterior value because most of the likelihoods are zero or near zero. This resulted in the posterior value being a minimum value for most of the locations. Then the (usually) single node that has a likelihood greater than zero dominates the results, causing the maximum a posterior location to be equal to the maximum likelihood location. Because of this, the results from this classifier are not discussed in any detail, but they are provided in all the tables of results.

Table 4.4 gives the locations visited for each test run. Test one was the shortest, visiting ten locations: five office locations and five hallway locations. Test two was a longer test, visiting 18 nodes: seven of those being office locations. This test run also visited the outdoor location, Patio. The third test was the longest, visiting every node in the map: 33 locations total, 14 office locations. The locations for the last two tests are less important, but it is easy to tell the route that was taken before and after the robot was 'teleported.' The asterisk represents the

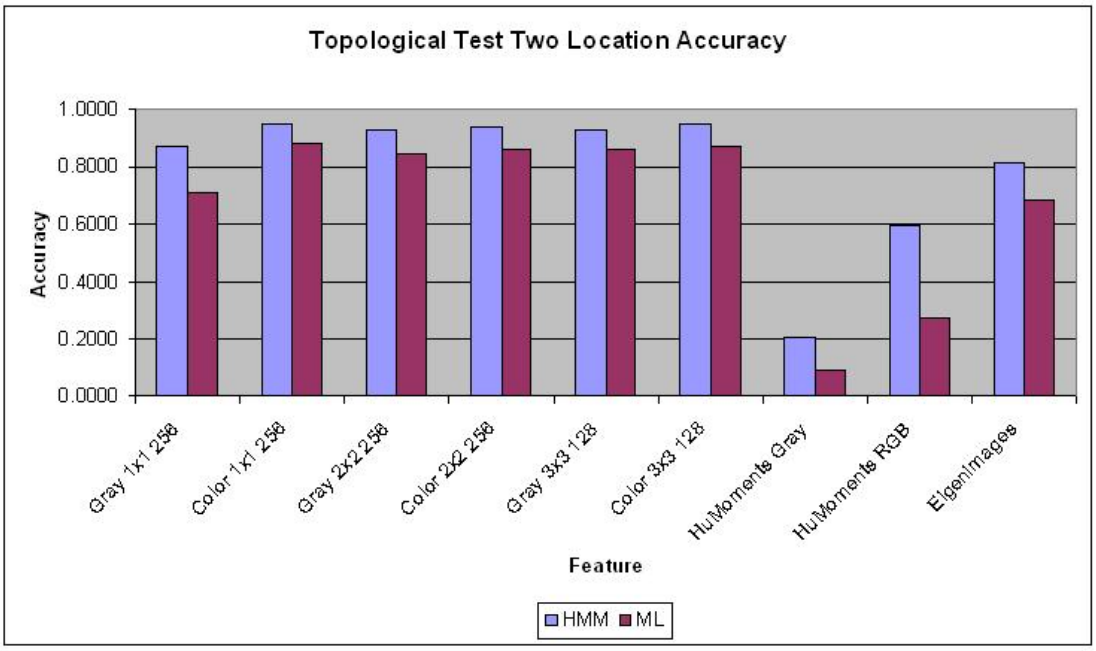

Figure 4.15: Topological test two: position accuracy

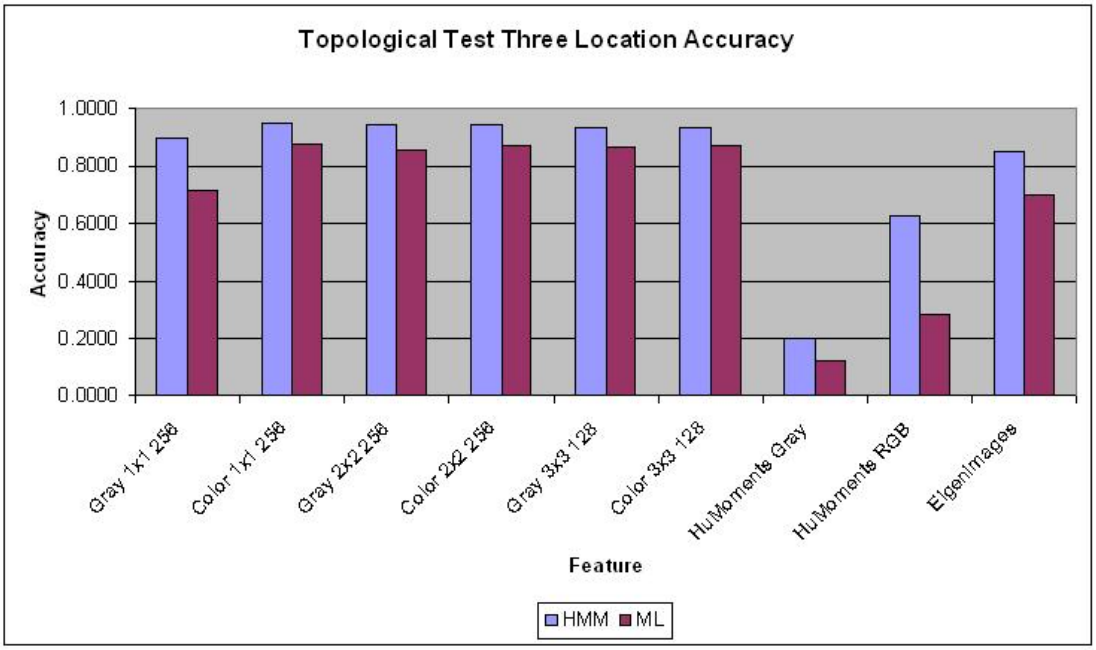

Figure 4.16: Topological test three: position accuracy

location where the robot was teleported. The teleported node was far away from its previous node so that the system would not have any substantial probability of moving from its previous location to the teleported location. This is important because the system contains higher probabilities for moving to locations that are closer to the current location. Figure 4.1 includes a reference of the locations visited.

Tables A.1 to A.8 include the complete results of these tests. All the results are given in units of *number of images or features*. For example, the Best Case column for the Time to Localize section gives the least number of images out of all ten tests to correctly localize the system after startup. Figures 4.14 to 4.16 summarize the results for the location accuracy for the first three tests. The results shown are the average of all ten tests.

All three tests showed that the HMM proved to give the best results, with several classifying over 92% correctly. The Hu Moments (Gray) classifier proved to be inadequate for these tests, and the Hu Moments (Color) feature was much better, but still inadequate to perform localization. The HMM tested better than the ML in every case.

The best results for Test one are obtained using the Color Histogram 3x3 feature which classified over 95% correctly. However, this result was only slightly better than those from several other features including: Color Histogram 1x1, Color Histogram 2x2, Gray Histogram 3x3, and Gray Histogram 2x2. All of these provided excellent results. The best results for the ML (Maximum Likelihood) classifier for this test are from the Color Histogram 1x1 feature, correctly classifying over 88% or the images correctly. Again, several others had results similar to that of the Color Histogram 1x1: Color Histogram 3x3, Gray Historam 3x3, and Color Histogram 2x2.

The best results from test two resulted from using the Color Histogram 1x1, 95%, and Color Histogram 2x2, 87%, for the HMM and ML, respectively. Test three best results are from the Color Histogram 1x1 feature for both HMM, 95%, and ML, 88%.

The Color and Gray Histograms for these three tests performed better than the other features, with the Gray 1x1 feature performing the worst of these. In most cases, the Color Histogram outperformes its corresponding Gray Histogram, however, the results are usually too close to be well differentiated by these tests. Several of the results show that many of the Color and Gray Histogram features classify over 95% correctly. As a result, a single feature cannot be chosen as giving the best results overall. The results do show that the histogram features do perform adequately for classifying the indoor locations.

The HMM model still requires that the ML classify a sufficient number of the locations correctly in order to perform adequately. The greatest improvement on the ML results is from the Hu Moments RGB feature in Test three where the ML classifies over 28% correctly and the HMM classifies over 62% correctly. When the ML jumps around, the nodes with the most connections tend to gain the higher probabilities. However, the Color and Gray Histogram features are usually better than its corresponding ML results by around 10%, which is significantly higher.

The ML classifier solves the global localization problem faster than the HMM as expected, because of its lack of 'memory.' However, the HMM was usually only somewhat worse than the ML, and was still very fast. Figures 4.17 to 4.19 show the summary of the localization time results. The time is measures in number of images. Because the experiments were performed offline, and the images can be captured at different rates, time in seconds does not give an accurate assessment. The number of images needed by the system gives an accurate assessment of how

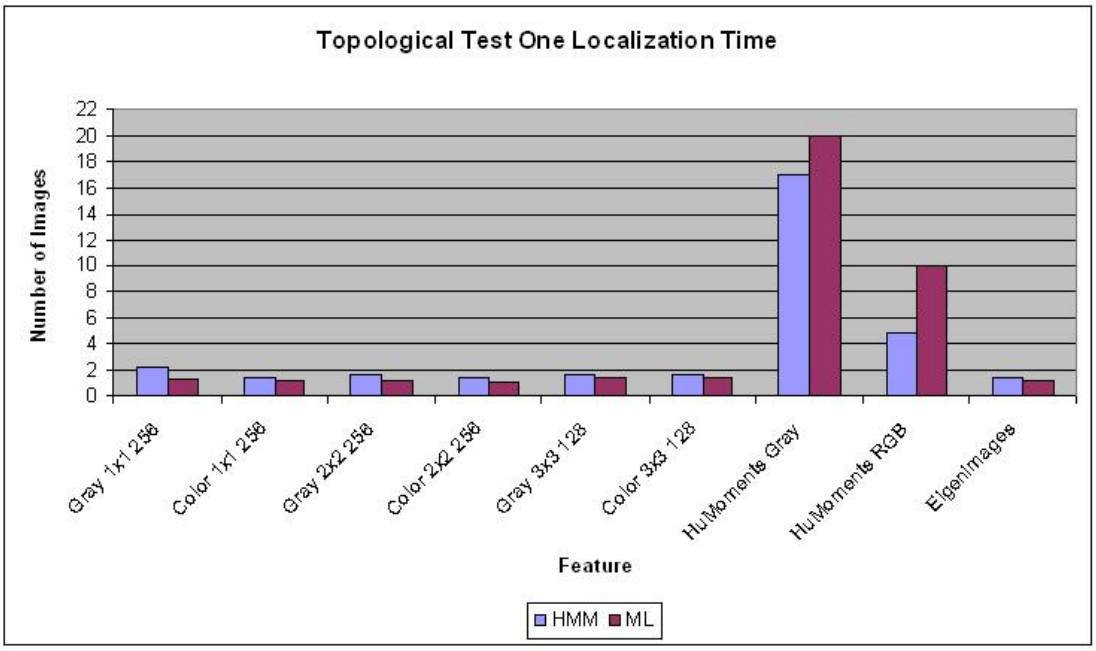

Figure 4.17: Topological test one: localization time

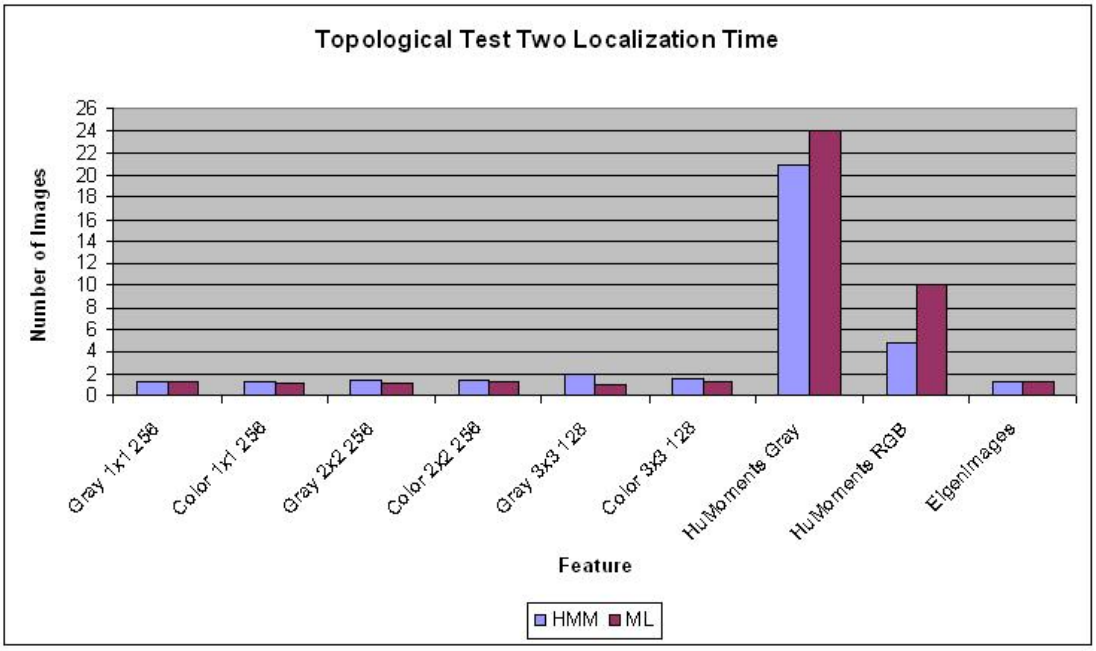

Figure 4.18: Topological test two: localization time

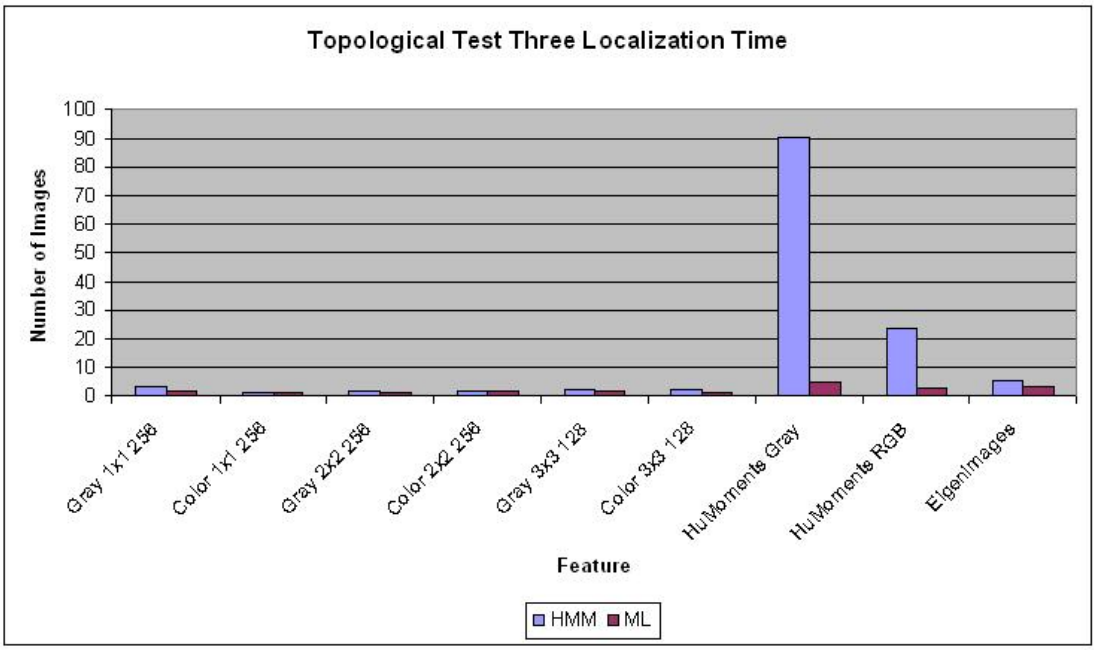

Figure 4.19: Topological test three: localization time

long it takes to localize the system. With the exception of the Hu Moments features, these tests show that the global localization time is sufficient for use on an autonomous mobile robot, usually requiring only two to three images before the system has determined where it is in the map. Tables A.2, A.4, and A.6 show the complete results for these three tests.

The Kidnapped Robot Problem tests (tests four and five), showed that again, because of its lack of memory, ML performed the best, as was expected, and that the HMM was not very far behind. Figures 4.20 and 4.21 show the results of the kidnapped robot tests. The results show the number of images, like the global localization results, required to relocalize after being teleported to another location. The results for all but the Hu Moments features show very good performance. The Hu Moments features again performed inadequately.

Tests six and seven, were performed in the second topological map, Figure 4.2.

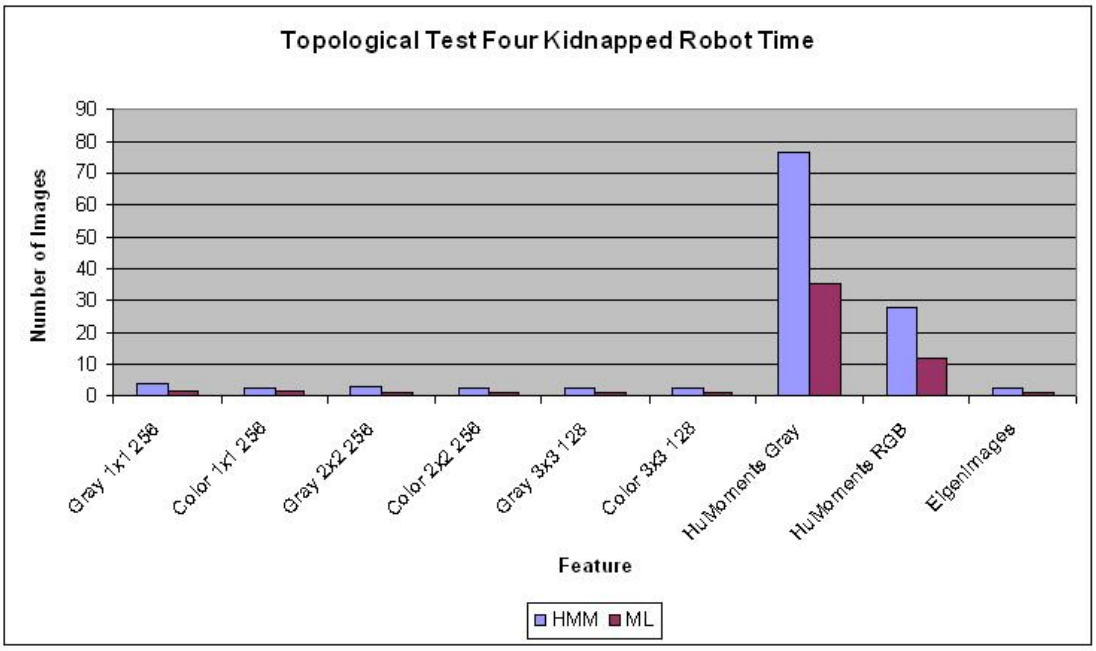

Figure 4.20: Topological test four: kidnapped robot problem

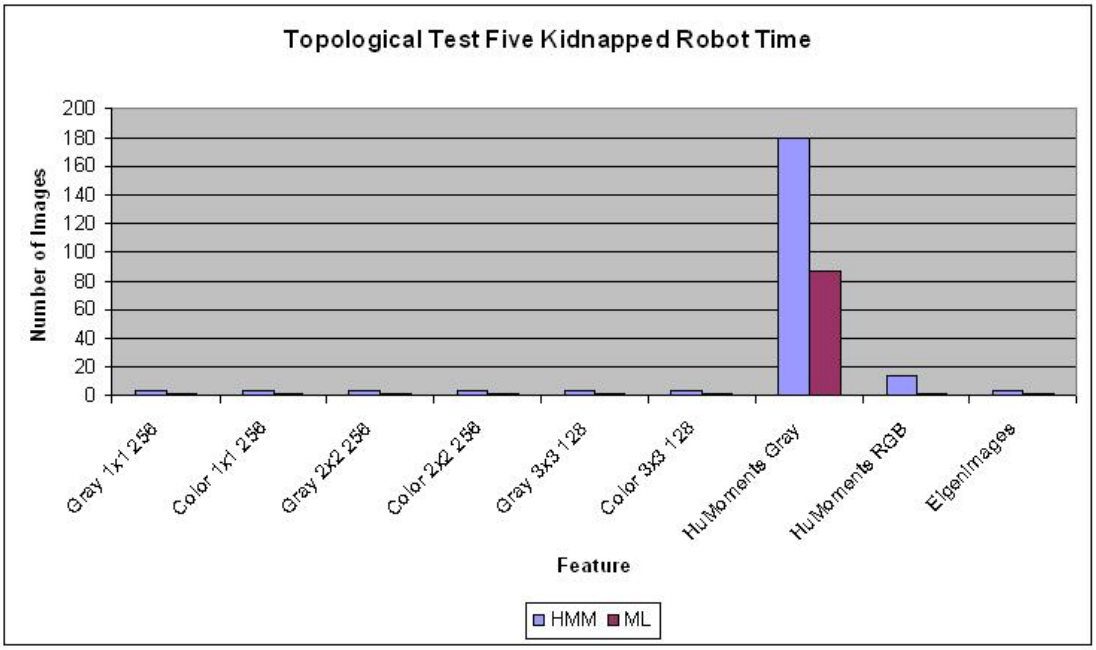

Figure 4.21: Topological test five: kidnapped robot problem

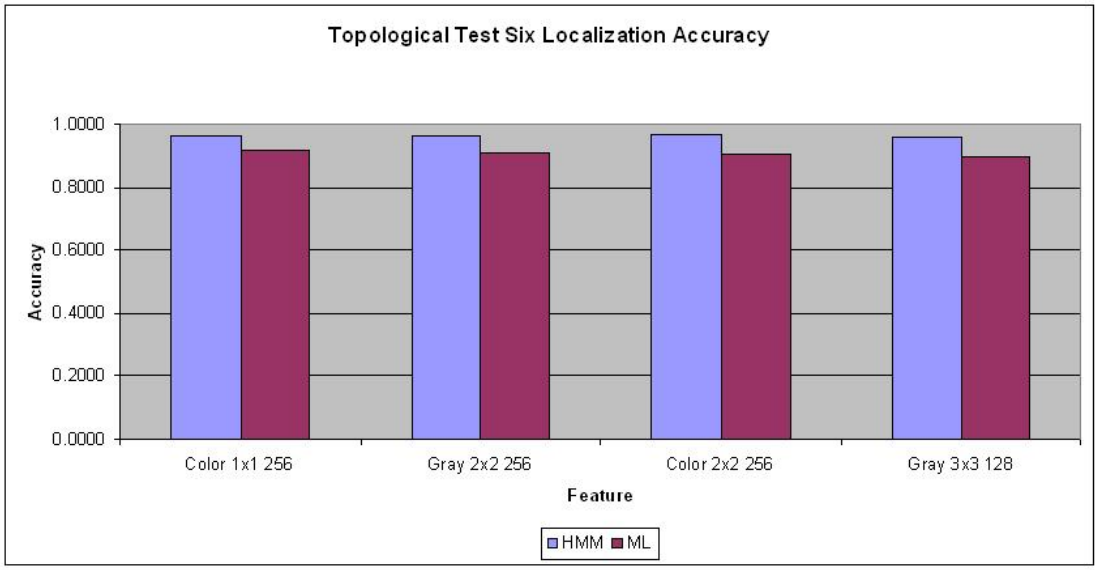

Figure 4.22: Topological test six: position accuracy

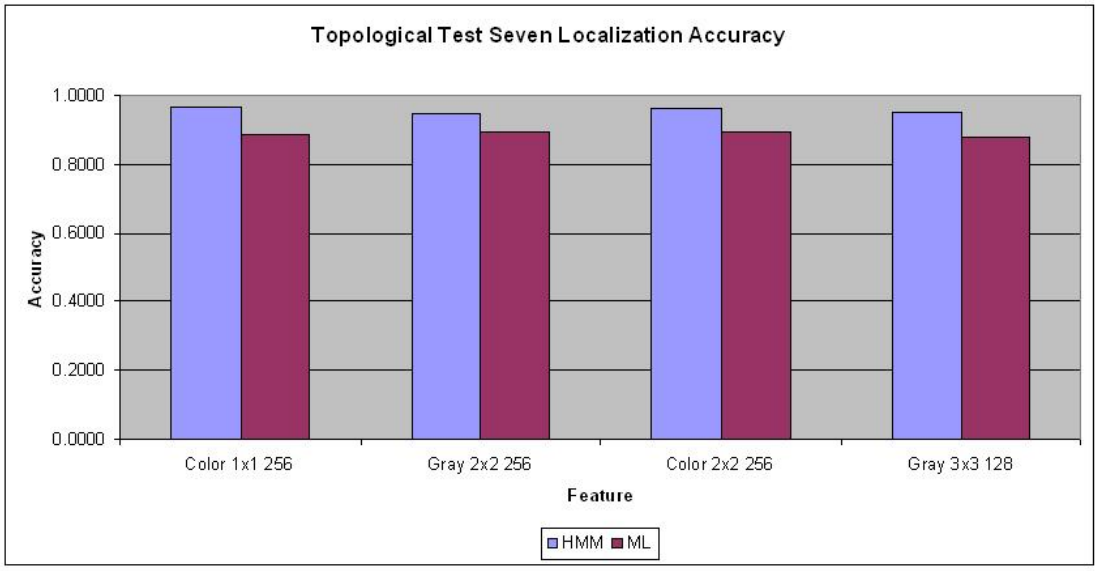

Figure 4.23: Topological test seven: position accuracy

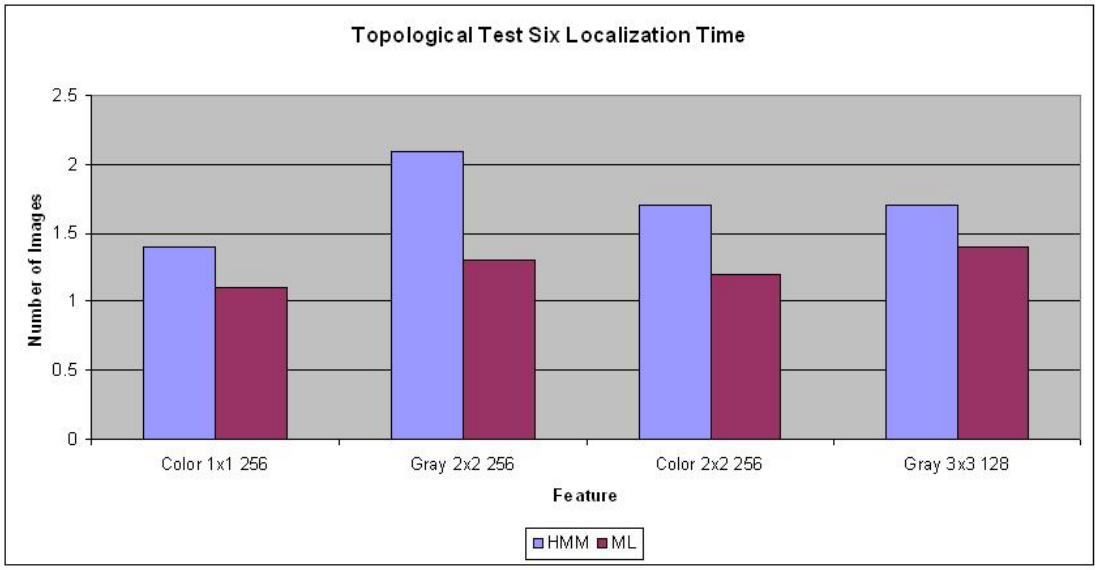

Figure 4.24: Topological test six: localization time

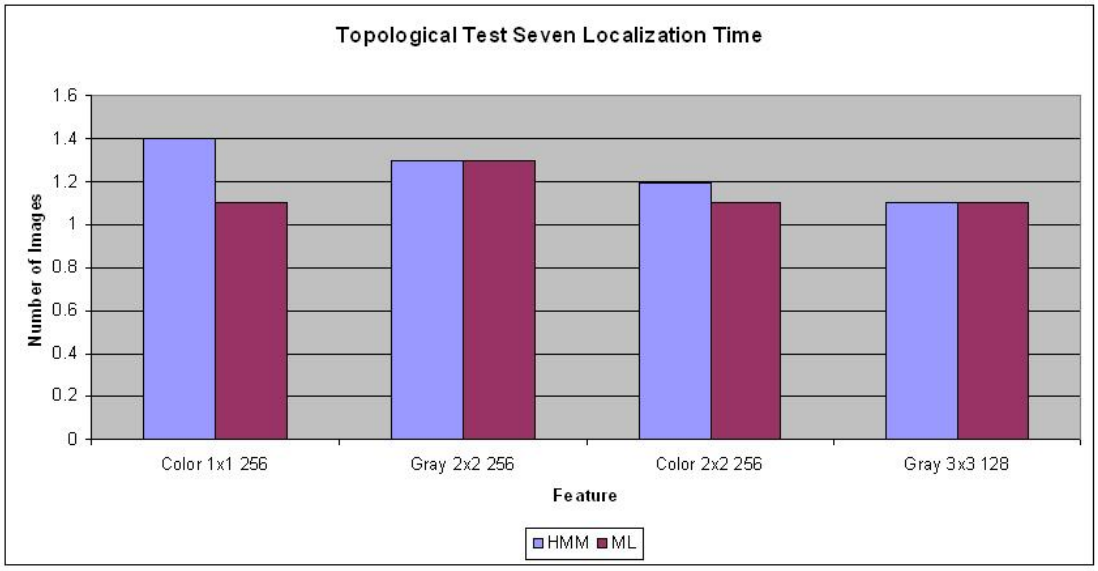

Figure 4.25: Topological test seven: localization time

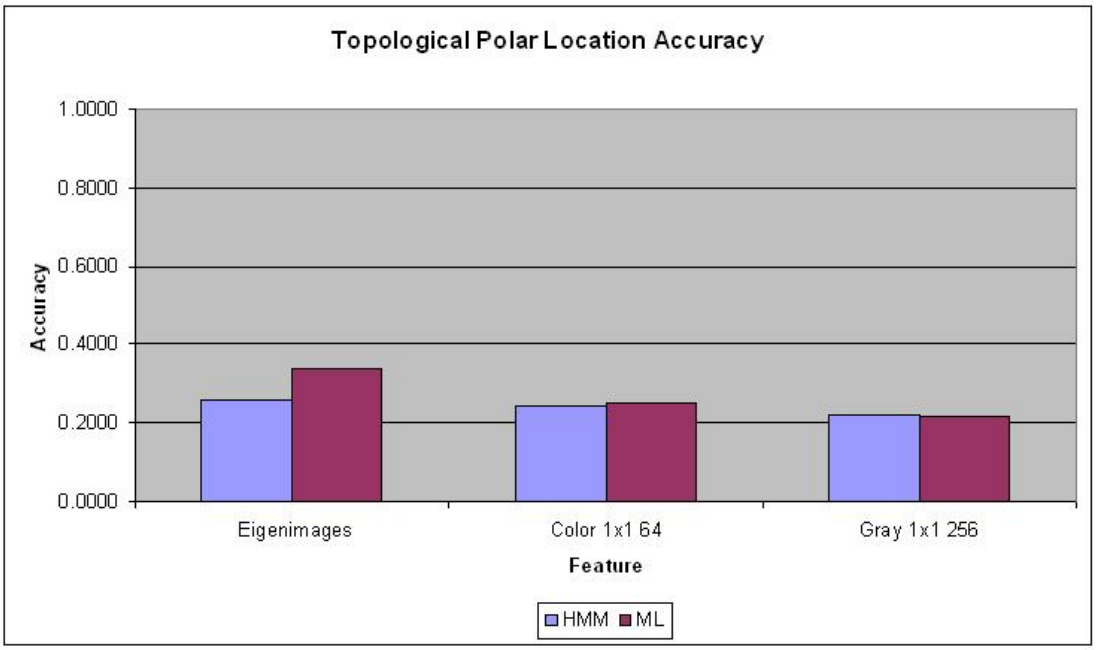

Figure 4.26: Topological polar test: position accuracy

These were meant to test how well the system performs outdoors. Test six visited every node in the map, similar to Test three for the other map. Test seven visited only the outside locations in the map, visiting all but one of them twice. The number of features used in these tests were reduced to the ones that performed the best in the previous tests. The features used were all histogram features, numbers 2, 3, 4, and 5 listed in Figure 4.3.

Both of these tests performed similar to those of the previous tests. Figures 4.22 to 4.25 show the summary of the results. Each feature was classified correctly by the HMM over 95% of the time for Test six, and over 94% for Test seven. The localization times for both tests were similar to those of the other map as well. Tables A.9 to A.12 provide the complete results for these tests.

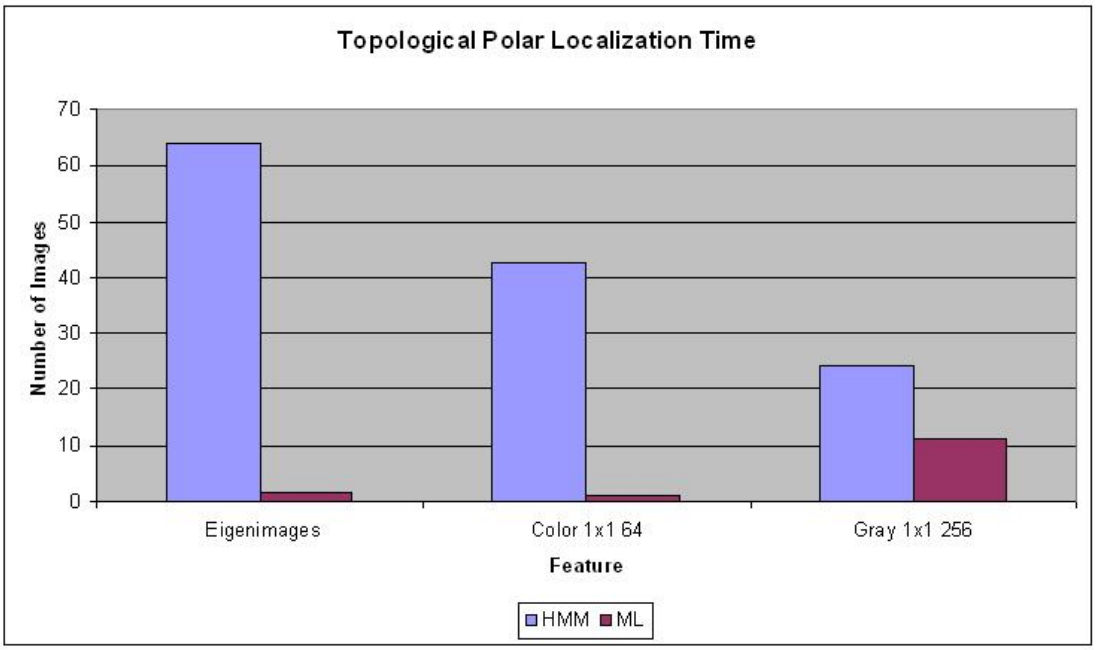

Figure 4.27: Topological polar test: localization time

4.2.2 Polar Testing

The testing of the polar images did not provide good results. Only two classifiers were used, the HMM and ML. Both classifiers were not able to localize with respect to any of the six locations. The results from these tests are shown in Figures 4.26 - 4.27. The full results are shown in Tables A.13 and A.14.

The results show that the system worked about 25% of the time. Sometimes the system was locked into one location, and other times the system varied greatly on the location that it had chosen. Most occasions, the system did not show any consistency or ability to process this map.

The classifier's performance is due to several factors. First, there were small number of images for each location. Previous tests had shown the best performance when the number of images representing each location was closer to 300. Second, because of the few locations and few images, not all of the descriptors

could be used. Using only 64 bins drastically reduces the amount of information. Previous testing had also shown that these descriptors with using fewer than 128 bins per histogram did not perform well. The best results were obtained from using as much information as possible in a feature descriptor, and then reducing the dimensionality using PCA. The last factor is how the pictures were taken and when. Many of the pictures were taken on different days in different conditions at different positions. This did not allow for modeling the images accurately using the GMM because most of the images varied significantly. This exacerbates the problem of a small number of images because the number of Gaussians in the GMM is very limited, compared to the other tests.

4.3 Discussion

The experimental results from the tests illustrate that the appearance-based localization method is a viable approach. The method works extremely well, at times over 95% of the time, on these tests. These tests also show the ability of the system to recognize several locations that look very similar. The hallways, for example, Figures 4.5 and 4.6, where the tests were performed are extremely similar, in color, size, and structure. The HMM proved to work very well for this system, improving the ML results by around 10%. These results show this system to have merits, in both indoor and outdoor environments.

The features used in these tests were not complex and can be calculated quickly, allowing this system to run in real time. However, as stated previously, all the tests were done offline in order to simplify evaluation and allow for multiple tests to be performed using the same images.

The polar testing was inconclusive because of the limited number of images.

Given a proper map, the results should greatly improve.

Limitations The proposed approach was evaluated using a map with 26 locations. A larger map will be needed to get a better determination of how well the system works for larger areas such as a college campus environment. The system also needs to be tested on maps of base camps in Antarctica and Greenland again, as there were not enough images of these camps to get sufficient results.

Another limitation is that the system was not tested under significantly varying lighting condition or other noise. The lighting condition inside the building does not change much, and the outdoor locations were all imaged under the same lighting conditions. Also, there was never a large crowd of people at the time the images were being captured. Of course, images that consist of only people could potentially cause the system to lose its position.

The database has not yet been built for multiple lighting conditions, and the results from how well the system works under the varying lighting conditions would be of interest. The lighting conditions in Greenland and Antarctica do vary as well from cloud cover and the direction of the light. As implied however, the simplest solution to varying lighting conditions is to capture images for the training set at different times of the day.

The system also relied on a single feature to localize. This was sufficient for the environments tested, but other environments may require a combination of weighted features or different features altogether. Not all features could be tested, and indeed there exist many more than were described in this work.

Future Work The future work will be based on testing the system in Greenland and Antarctica and increasing the size of the map. In these environments, other

features may need to be tested in order to localize sufficiently. Also, the area must include some structure or texture that does not disappear after a short time. So the system in Antarctica and Greenland would be limited to within camps where there are some structures.

Chapter 5

Geometric Localization

This chapter describes the approach and results for performing localization on a geometric map. A geometric map would be stored in a node in the topological map. Once the topological system has determined which node the system is currently in, then the geometric map for that location is used to perform the geometric localization. Combined, these two steps yield a hybrid approach.

5.1 Approach

Geometric localization in this section is performed somewhat differently than the topological localization. This step utilizes a maximum likelihood method. Maximum likelihood was chosen because of the lack of prior knowledge. This is because the topological system must decide which geometric map is to be used and it can change from one image to the next. The amount of memory and data needed to store the states of a HMM for every node would probably be too large.

This step is also used to determine the orientation of the system which is meaningless in the topological map because the adjacency graph does not describe

distances, sizes, or any geometric information. Therefore, the goal of the geometric localization step is to determine both the position and orientation with a high accuracy.

Localization is based on matching images to a specific orientation within a specific location (grid) in the map. Scale Invariant Feature Transform (SIFT) keypoints are used to determine the matches. Images are taken at a specific location every 45 degrees for a total of eight different orientations (0, 45, 135, 180, 215, 270, and 315). The system determines within 45 degrees. All the images from these different orientations are used to represent the location as a whole.

Localization is performed by first matching a test image's keypoints to each location in the map. A match is determined by the location that had the most matches. The number of matches is used to represent the probability, with the higher the number of matches, the higher the probability. Therefore, the location is the one that had the most matches.

After a location has been determined, the matching is made against the keypoints that represent each orientation within the location. Again, the orientation that has the most matches is the orientation that is chosen as the match.

The keypoints are broken into two different databases. The first is a database of images that represent the locations in the map. The second is a database that represented each orientation within each location.

5.1.1 Scale Invariant Feature Transform

The scale invariant feature transform (SIFT) [41] [40] was the only feature used for geometric localization. SIFT is a local descriptor, meaning that the features

describe only a small part of the image, in contrast to global features that describe the entire image. The SIFT features are designed to be invariant to scale, rotation, and translation. In addition, they are partially invariant to illumination.

David Lowe, provided a program that generates SIFT keypoints from an image [42]. This program was used to generate the features used in these experiments. A SIFT keypoint is made up of a location (x,y), scale, and orientation, and a 128 element feature vector. These are generated at locations in an image of local maxima and minima in the scale space. The scale space of an image is produced by convolving the image with a Gaussian kernel. The maxima and minima are found by subtracting the values from adjacent scale spaces [41].

The SIFT program generates many features from a single gray-scale image, varying based on the image and its size. Typically, around 300 keypoints were generated for the indoor images used in the these experiments, and 500 or more for the outdoor images. The outdoor images contained more texture than the indoor, resulting in more keypoints.

[46] describes and tests several different local descriptors including SIFT. The descriptors were tested on their matching ability, and the experimental results showed that the SIFT features and variants of the SIFT descriptor produced the best results. Because of how well it tested and the availability of the SIFT program, it was the feature used for the experiments.

Matching SIFT Keypoints

As described in [41], a cluster of three or more keypoints is best for producing accurate matches. Therefore, each keypoint is clustered based on its location, orientation, and scale. The keypoints are grouped using a 4D space (x, y, orientation, scale) to group all the keypoints. Once a keypoint has been placed

in a specific bin, it is also placed in every one of its neighboring bins. This allows for a wider range of what constitutes up cluster.

A match is determined by calculating the Euclidian distance between the 128 element descriptors of two keypoints. If the distance of the closest match is much smaller than the second closest match, the two keypoints are said to be a match. In order to be considered a match for the entire image however, three or more keypoints in the same cluster that the matched keypoint is in must match the keypoints in the same bin. For example, if there are three keypoints in cluster A in image 1, and the first keypoint matches cluster B in image 2, then the other keypoints in cluster A must all match to cluster B as well in order to be considered a match. This procedure is described in [40].

5.1.2 Geometric Maps

Three different locations were used for experiments. Each location is a 4x5 grid with eight possible orientations from each location (every 45 degrees). Figure 5.1 shows the layout of each of the geometric maps and the orientations at which the pictures were taken. Two of the maps are used from an indoor location and one from an outdoor location. The indoor maps were generated at the same location of two different sizes: the first at three feet intervals (12ft x 9ft), the second at one foot intervals (3ft x 4ft). The origin for both of the indoor maps were at the exact same location, so the indoor maps are of the same area, but one overlaps the other and encompasses a larger area. The reason for the different sizes on the indoor map is to determine how accurate the map can be at the different sizes, and if there is a change accuracy as the interval size becomes smaller. The outdoor map was created at one foot intervals (3ft x 4ft).

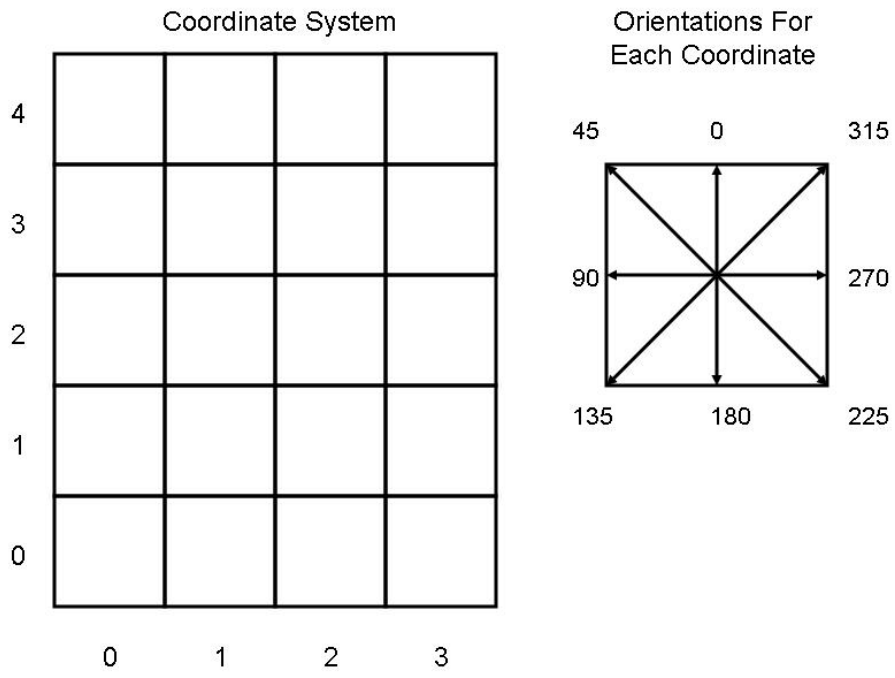

Figure 5.1: Layout of the geometric maps

Figure 5.2: Two adjacent locations of the one foot interval indoor map. Top: 0 degrees at (0,0). Bottom: 0 degrees at (1,0)

Figure 5.3: Two adjacent locations of the one foot interval indoor map. Top: 45 degrees at (0,0). Bottom: 45 degrees at (1,0)

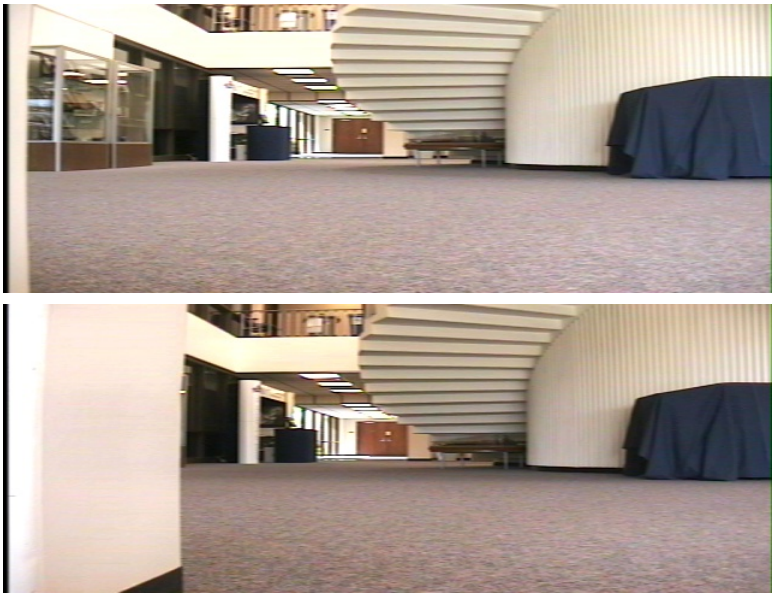

Figure 5.4: Two adjacent locations of the one foot interval indoor map. Top: 90 degrees at (0,0). Bottom: 90 degrees at (1,0)

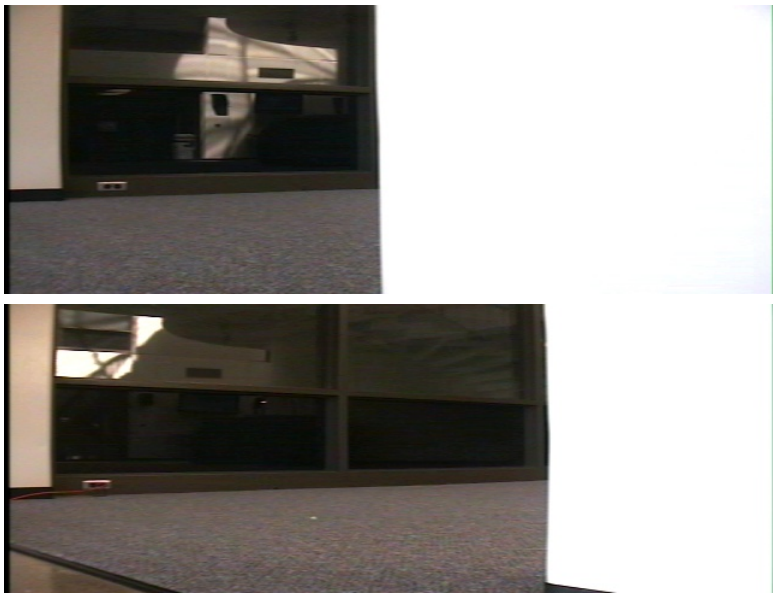

Figure 5.5: Two adjacent locations of the one foot interval indoor map. Top: 135 degrees at (0,0). Bottom: 135 degrees at (1,0)

Figure 5.6: Two adjacent locations of the one foot interval indoor map. Top: 180 degrees at (0,0). Bottom: 180 degrees at (1,0)

Figure 5.7: Two adjacent locations of the one foot interval indoor map. Top: 225 degrees at (0,0). Bottom: 225 degrees at (1,0)

Figure 5.8: Two adjacent locations of the one foot interval indoor map. Top: 270 degrees at (0,0). Bottom: 270 degrees at (1,0)

Figure 5.9: Two adjacent locations of the one foot interval indoor map. Top: 315 degrees at (0,0). Bottom: 315 degrees at (1,0)

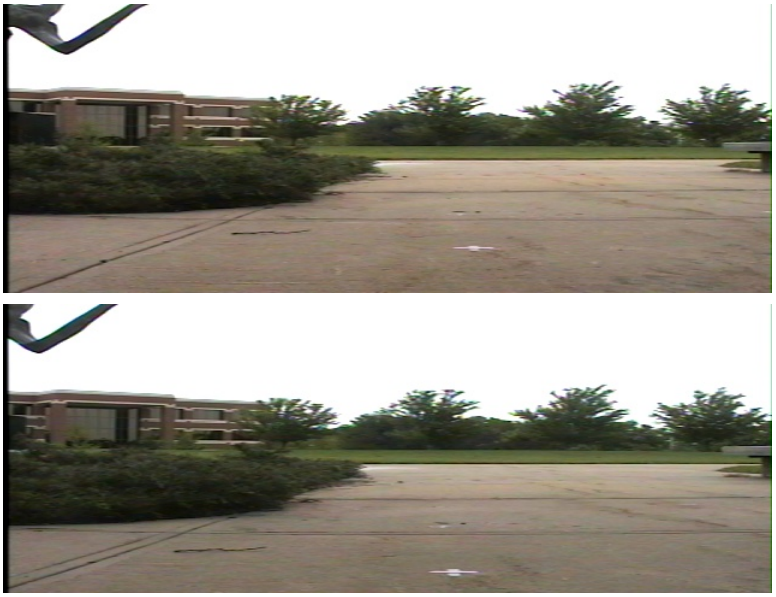

Figure 5.10: Two adjacent locations of the one foot interval outdoor patio map. Top: 0 degrees at (0,0). Bottom: 0 degrees at (0,1)

Figure 5.11: Two adjacent locations of the one foot interval outdoor patio map. Top: 45 degrees at (0,0). Bottom: 45 degrees at (0,1)

Figure 5.12: Two adjacent locations of the one foot interval outdoor patio map. Top: 90 degrees at (0,0). Bottom: 90 degrees at (0,1)

Figure 5.13: Two adjacent locations of the one foot interval outdoor patio map. Top: 135 degrees at (0,0). Bottom: 135 degrees at (0,1)

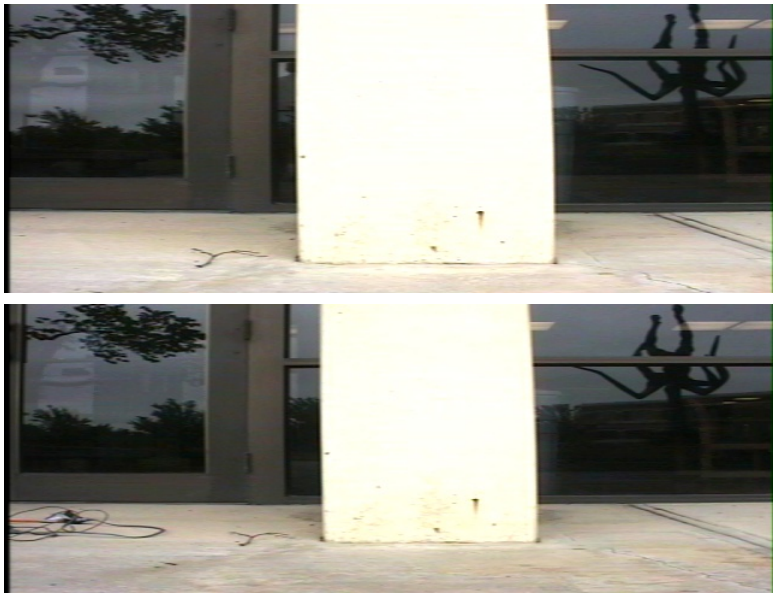

Figure 5.14: Two adjacent locations of the one foot interval outdoor patio map. Top: 180 degrees at (0,0). Bottom: 180 degrees at (0,1)

Figure 5.15: Two adjacent locations of the one foot interval outdoor patio map. Top: 225 degrees at (0,0). Bottom: 225 degrees at (0,1)

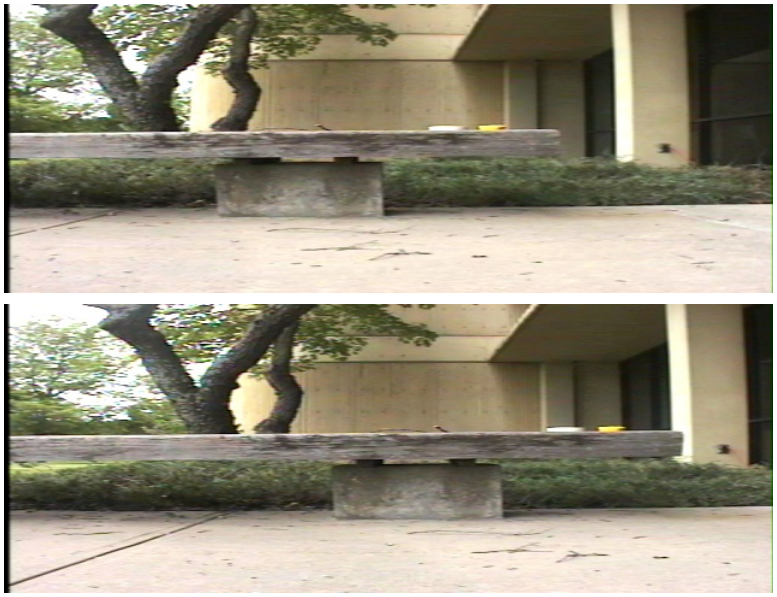

Figure 5.16: Two adjacent locations of the one foot interval outdoor patio map. Top: 270 degrees at (0,0). Bottom: 270 degrees at (0,1)

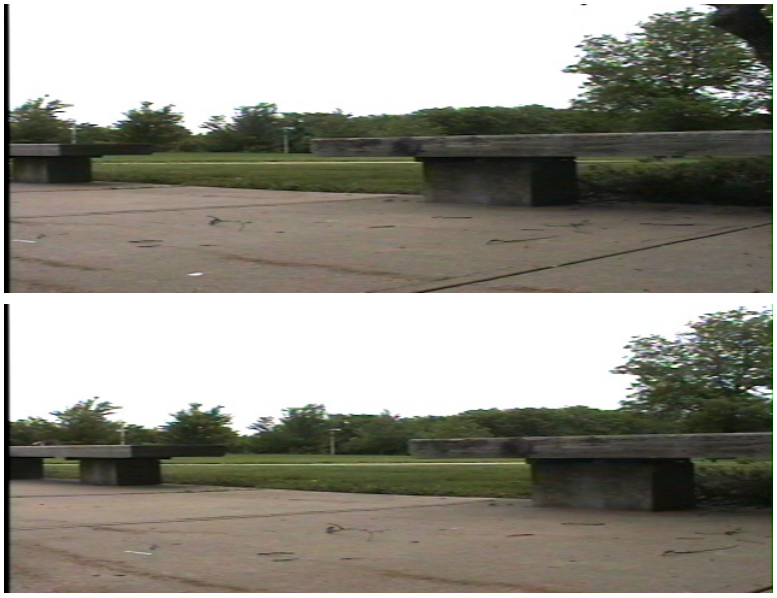

Figure 5.17: Two adjacent locations of the one foot interval outdoor patio map. Top: 315 degrees at (0,0). Bottom: 315 degrees at (0,1)

The images for the maps were acquired by using a robotic base platform under remote control. A grid was set up at the location where the images were taken. The robot was set in the middle of each grid and it captured 20 images in each of the eight directions. Figures 5.2 to 5.9 show the images that were used for locations (0,0) and (1,0) for the one foot interval indoor map. The locations are adjacent to each other and the figures show how similar the images are. Similarly, Figures 5.17 to 5.17 shows images from the outdoor patio map at every orientation for locations (0,0) and (0,1).

This process generated 160 images for each location. 10% of the images were selected to be used as part of the test images; and 144 of the images were used as part of the training set. Some of the training images and test images did contain pictures of people, but very few.

Not all keypoints generated from an image were necessarily used to represent a specific location. The percentage of keypoints generated from an image that

matched the database prior to being placed in the database was determined. If this percentage was larger than some preset threshold, then the keypoints for that image would be placed in the database. This was done for both the location and orientation databases. The values used in this test were 20% for the location, and 80% for the orientation. This allowed for a higher percentage of keypoints for each orientation, and also attempts to limit the number of keypoints for each location, which usually have a high similarity. But because the number of images to represent a specific orientation within a specific lcoation were much smaller than for a location, the number of keypoints used to represent the orientation was not very large.

The number of keypoints used to represent a location in the indoor maps usually around 3,000. The number was in the hundreds for representing each location. The number of keypoints in the outdoor maps was around 8,000 because each image contained many more keypoints. Matching against 3,000 keypoints can be done fairly quickly, but matching 8,000 keypoints takes too much time to process in real time.

Two different databases were created for each map for testing purposes. The first database used the images as they were captured from the camera. The images usually did not contain much variability unless a person walked into the sight of the camera, but each image did generate different keypoints. The second database was the same map with Gaussian noise with a variance of 0.003 added to every image. This insured that every image differed by a greater amount than the other database. These databases with the added Gaussian noise were treated as an entirely different map, allowing for six different maps to be tested in the system.

5.2 Evaluation

The tenfold testing procedure was used to evaluate the system. The six databases created were each tested using the same procedure.

For each test, every test image in the test set was matched against its corresponding training set database; and the results were compared to its actual orientation and location. The number of correct locations and orientations were evaluated separately.

Figure 5.18 shows what one of the matches for orientation could look like. Each image includes two parts. The top is the test image that was not part of the training set and the bottom image is of an image that represents one of the orientations for a specific location in a the map. Lines are drawn to show where the matches occur to give an idea of how the system works. The keypoint matches in the images shown do not require the cluster of matches that were required in the geometric system, so there were fewer matches in the actual system.

Results The summary of experimental results for all six tests are shown in Figures 5.19 and 5.20. Table B.1 lists the complete results for each of the geometric tests.

The results show that the classifier worked well for all maps in determining the location, including the indoor and outdoor tests. The outdoor tests worked extremely well, slightly better than the indoor tests. This is the result of the outdoor images containing more keypoints. This also caused the outdoor tests to run more slowly than the indoor tests.

In all tests, the correct location was chosen over 90% of the time with the best being the outdoor tests at over 98%. The worst results came from the databases

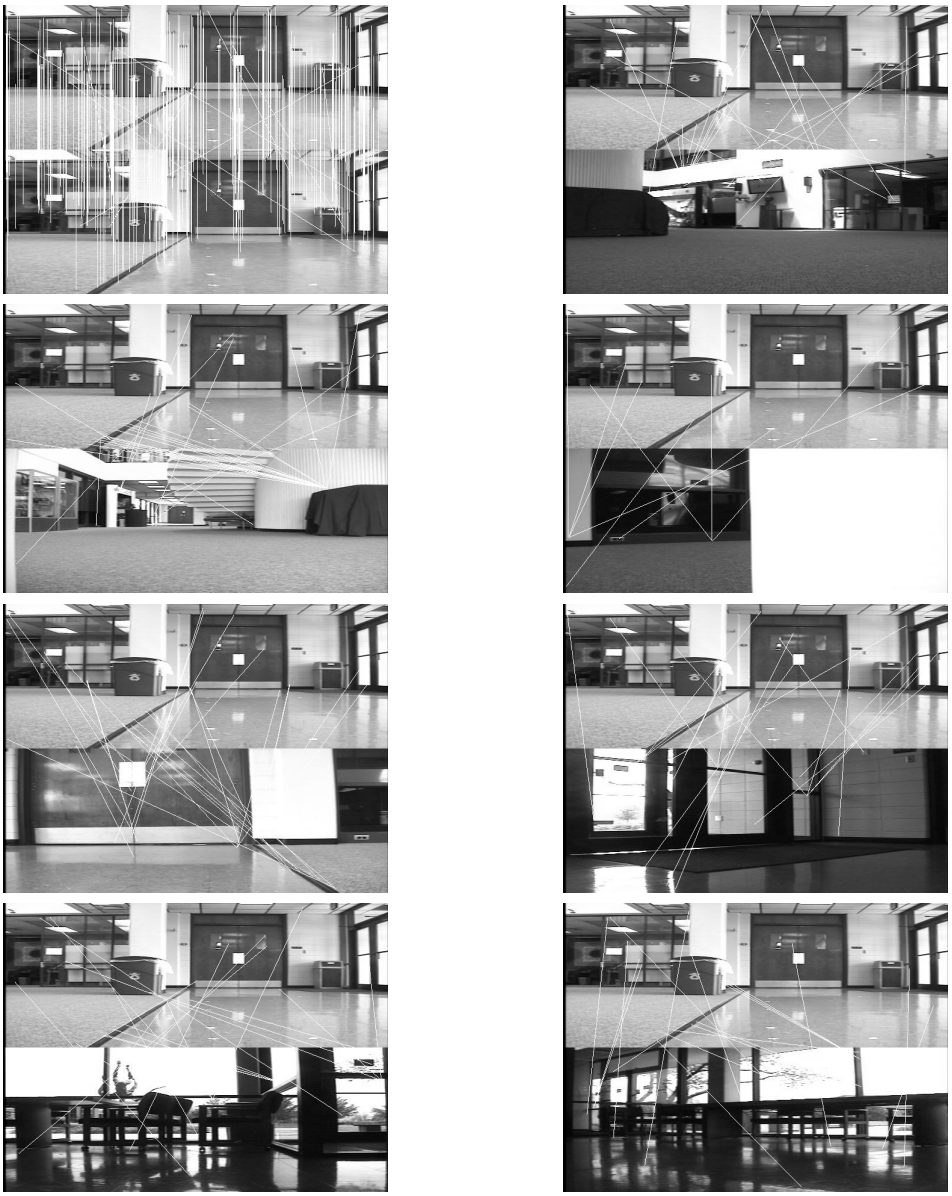

Figure 5.18: Geometric image matching. The sample images show how matches are made. The eight different images represent one image from each of the eight orientations (bottom) and the test image that was used to compare (top). Lines are drawn between matching keypoints. The number of lines represents the number of matches. The top left image, representing the image at zero degrees, matched the best in this case.

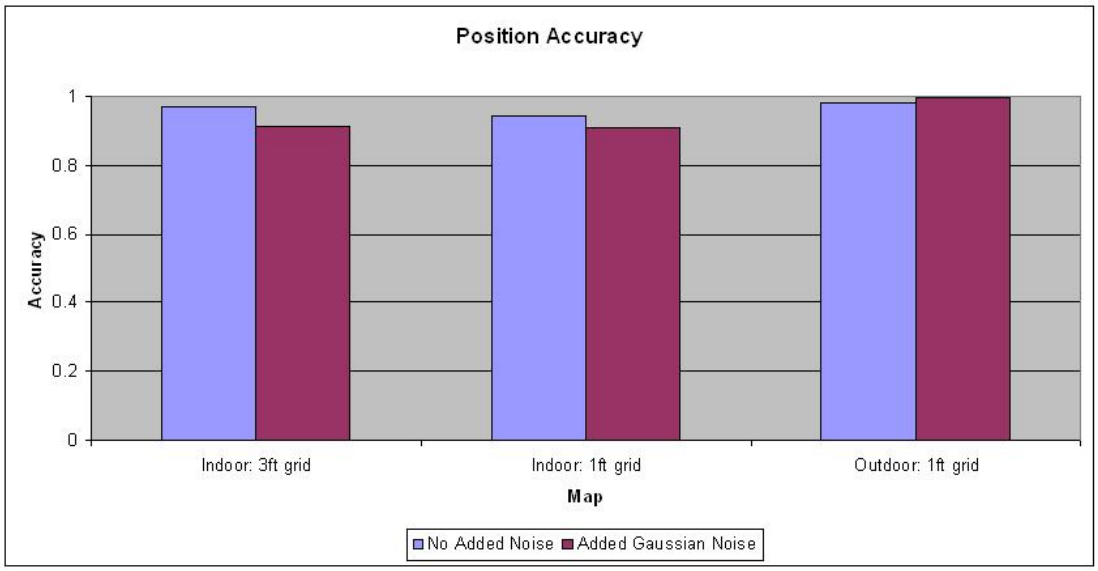

Figure 5.19: Geometric position test results

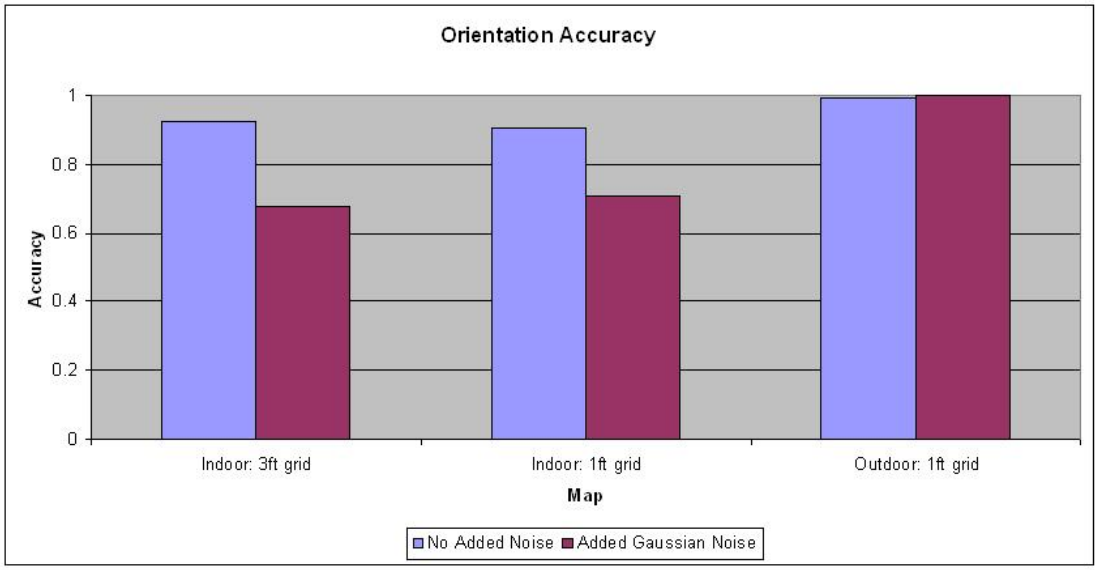

Figure 5.20: Geometric orientation test results

with the Gaussian noise added to each image where the indoor map at one foot intervals determined the correct location 90% of the time. This number is well within range to allow a robot to navigate properly.

The accuracy of determining the orientation was usually less than that of determining the location. This happened because there were less images used to represent each orientation and therefore fewer keypoint clusters. The other reason was because of how the images were selected to represent the location. Numerous keypoints were not considered that should have been used to represent the orientation. A better method of selecting specific keypoint clusters from each image instead of selecting all or none could be used instead.

The best results of choosing the proper orientation was from the outdoor experiments. The result of the outdoor tests containing more keypoints than the indoor tests. The indoor tests with noise added performed the worst. The added noise tends to be processed as a new keypoint. The noise causes many of those keypoints to be more similar than they were in the images without the noise. Because the indoor images had far fewer keypoints than the outdoor images, this caused the orientations to look more similar. Too many keypoints were selected that were generated from noise in the image. This could be resolved by smoothing the image before generating the keypoints or ignoring the fact that there is noise and choosing a more representative sample of keypoints. However, even with the noise, the indoor orientations were chosen correctly 67% and 70% of the time. In most cases, when the orientation was incorrect, it was off by 45 degrees, the smallest value given the interval of degrees used in the database. The position accuracy was not affected as much because of the larger number of keypoints that were used to represent a location versus the orientation. This is again because of how keypoints were selected.

The speed of each comparison was not being measured, but the results were slower than could be performed in real-time. This is because the system was comparing every single location when it did not have to. The speed could be improved by using a quad tree for the map, and selectively using keypoints to test for at each location in the quad tree. Higher levels on the quad tree would contain fewer keypoints per location than those on lower levels. This would allow the system to throw out several locations without explicitly checking every keypoint at those locations, as long as the keypoints chosen were representative of the locations.

5.3 Discussion

The appearance-based localization process was continued on from the topological testing into the geometric experiments. The difference is the features that are used. The topological system used single features to represent the entire image, whereas the geometric feature was a set of keypoint clusters. The time to compare matches is increased, but the results are similar.

The geometric localization system worked well in all of the tests, with the orientation accuracy being reduced in the indoor tests with the added Gaussian noise. If the simple improvements described earlier are used, this system could probably work at around 10-15 frames per second. Again, exact testing of the time was not measured because the system was running offline, loading all the keypoints and databases from the disk for each comparison.

It is also shown that the system can work effectively on the scale of one foot intervals for location and 45 degree intervals for orientation. This should be sufficient for most applications. However, if a system does not require such

accuracy, the three feet interval would create a smaller database of an area and give somewhat better results because the images at the three feet intervals vary by a greater amount.

Limitations The system performs well at one foot intervals for location and 45 degree intervals for orientation, but some systems might need more accuracy such as a small vacuum cleaner robot that is smaller than 1 square foot. The system should be tested to see what size of locations it can correctly differentiate. The change in accuracy of moving from a three feet grid to a one foot grid dropped from 97% to 94%. If building a map at six inch intervals performs similarly, then the system might be able to perform at around 90% or better. But further testing needs to be done to determine if this is the case.

As described earlier, the system as is, is too slow to work real time. The performance needs to be tested in an online manner in order to determine exactly the speed it runs at and how best to make improvements.

Building the map of the area is a very time consuming process. The 4x5 grids used in this system took about 2 hours to get all the required images. This is because the system had to be placed in the proper orientation and location for each position, and then the camera can capture the images it needs. If the system were to be tested on even finer scale maps, the time to take the images would increase.

A system must be built that can automatically capture the necessary images. Doing this outdoors might be more difficult than indoors however, as the boundaries of each location in the map are not necessarily well defined.

Future Work More work needs to be done in the following areas: reducing the number of keypoints used to describe a location or orientation, testing a finer degree of orientations and a smaller grid, and speeding up the matching process by using a quad tree.

Chapter 6

Conclusion

The system, together with the topological localization and geometric localization, were found to be sufficient to localize a system over a large-scale environment with one foot and 45 degrees of its actual pose. This system should be sufficient for most mobile robotic applications.

The system, however, has not yet been tested enough to determine how well it works in inclement weather or at different times of the year. If the changing season is a problem, then the maps need to be created with images from different times of the year.

Building the geometric map is too cumbersome to do over a large area however, and this process must be automated using SLAM techniques. A robot could probably build the map at a faster rate than a human.

The topological localization system does require a great number of images in order to work adequately. Tests revealed that around 200 to 300 images were needed per location in order to perform well. The system will also have difficulties if the images of the different locations are not representative of the locations. The simple features used for this allows the system to classify locations quickly.

The hidden Markov model proved to work very well, even in cases where the maximum likelihood of the images did not match to the correct location. The added state information allows the system to globally localize in a short amount of time as well as re-localize when the robot becomes lost. This is an important task for an autonomous mobile robot.

6.1 Contributions

The contributions of this research is in designing a global localization system that performs well in both indoor and outdoor environments over a large-scale environment using only a single camera. The system can be stand alone, or an addition to an already existing system because of its simplicity and portability.

Many systems use a combination of sensors such as laser range finder, sonar, and stereo cameras. These devices can be cumbersome to a small robot or a person such as a mail carrier that need to use the system. The proposed system could also be easily added to an already existing system to increase reliability.

Most systems are designed for either indoor or outdoor environments, but not both. Outdoor only systems typically rely on GPS and indoor systems use the structure and the rich environment to be able to localize.

The other systems that have successfully performed localization in both environments have usually relied on position tracking methods where the initial location must be given to the robot. With the global localization system, the robot can be placed anywhere in the environment without knowing its initial location.

6.2 Limitations

There are a number of issues with using only a camera to perform localization. One is that localization can be impossible to determine for some images. For instance, an image that was taken close to a wall will not give enough information to localize, or the weather may prevent an image from matching well. There may be instances where the camera is not robust enough by itself to retrieve the necessary data to localize. In these instances, a multiple sensor system would be more robust. Another problem is that illumination changes or lack of light can cause the system to fail, whereas a range finder will continue to work in these conditions. Dynamic environments in general are problematic, but several methods have been used in the past for dealing with dynamic environments.

Not using odometry can cause the system to be less accurate than those systems that do use it. Odometry can allow a robot to localize itself to a finer scale than was used in this system. This is necessary though because odometry is not available in some of the target applications. It is not determined if using optical flow calculations will provide enough accuracy to replace odometry. However, [19] describes a correlation based optical flow algorithm used in a car for vehicle navigation. Because odometry is assumed to be inaccurate, an optical flow algorithm such as this might be able to perform well enough.

The proposed system is meant to work in Greenland and Antarctica, but because of the lack of texture, the system will be limited to locations around the campsite or around features that do not disappear after short periods of time.

6.3 Future Work

Work on localization has moved towards SLAM (Simultaneous Localization and Mapping) methodologies. Similarly, this work should move towards this as well. Also, mapping large-scale areas can pose a problem as there must be limitations on where the robot can go.

Gathering the images required for the topological localization system is not very time consuming, but gathering the images for the geometric maps does take significant amount of time. A system could be used where the topological map is initially made manually. Then a robot could traverse the map and build the geometric map for each location in the map. After this is completed, then any robot that traverses the location within the map could continually update the images in the map. This way, images that are representative of the location for all times of the year could be collected. This also would help for locations that are very dynamic. For instance, a new structure being built next to a location that is part of the map. This would allow the robots to not become confused by the new structure outside.

The testing of this system relied on a single camera, however multiple cameras can be used to increase the reliability, or using a panoramic camera instead. Both of these could add more features at each time-step which would help in the cases where one of the images is a close up of a wall or an image that has not been seen before.

6.4 PRISM/CReSIS Robotics

The localization system can also be used in a team based multiple robot scenario like the one described in [20]. The system could use the geometric localization to determine its location relative to the other robots. If a combination of sensors are used, for instance the lead robot uses a GPS to determine where it needs to be, then all the other robots could use the system to create a formation based on the location of the lead robot.

Currently, a UAV is being designed to help take radar measurements. The carrying capacity of the UAV is extremely limited. The load is mostly made up of the radar equipment and antennas. The UAV is currently remote controlled when landing or taking off. Adding a localization system could help it to locate and stay on the runway when trying to take off or land.

The PRISM/CReSIS robot [3] should be completely autonomous in the future. This requires the robot to be able to return to a polar camp to unload its data and to refuel. In order to do this without endangering the people in the camp, the robot will need a localization similar to the one described in this work to be able to find the locations to unload the data and refuel before returning to the field to take more measurements.

Bibliography

[1] E. L. Akers, "Modeling and Simulation of a Mobile Robot for Polar Environments," Master's thesis, University of Kansas, Department of Electrical Engineering and Computer Science, Lawrence, KS, October 2003.

[2] E. L. Akers and A. Agah, "Topological localization using appearance based recognition," *Submitted to the Journal of Robotics and Autonomous Systems*, 2007.

[3] E. L. Akers, H. P. Harmon, R. S. Stansbury, and A. Agah, "Design, fabrication, and evaluation of a mobile robot for polar environments," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. Anchorage, Alaska: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, September 2004, pp. 109–112.

[4] E. L. Akers, R. S. Stansbury, and A. Agah, "Long-term survival of polar mobile robots," in *Proceedings of the 4th International Conference on Computing, Communications and Control Technologies (CCCT)*, vol. II, Orlado, Florida, July 2006, pp. 329–333.

[5] E. L. Akers, R. S. Stansbury, A. Agah, and T. L. Akins, "Mobile robots for harsh environments: lessons learned from field experiments," in *Proceedings*

of the 11th International Symposium on Robotics and Applications (ISORA), Budapest, Hungary, July 2006, pp. 1–6.

[6] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *Robotics and Automation, IEEE Transactions on*, vol. 13, no. 2, pp. 251–263, April 1997.

[7] C. A. Bouman, "Cluster: An unsupervised algorithm for modeling Gaussian mixtures," April 1997, available from http://www.ece.purdue.edu/~bouman.

[8] W. Burgard, A. B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "The Interactive Museum Tour-Guide Robot," in *AAAI/IAAI*, 1998, pp. 11–18. [Online]. Available: citeseer.csail.mit.edu/burgard98interactive.html

[9] W. Burgard, A. Derr, D. Fox, and A. Cremers, "Integrating global position estimation and position tracking for mobile robots: the Dynamic Markov Localization approach," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '98)*, Victoria, BC, Canada, October 1998, pp. 730–735.

[10] D. Caltabiano, G. Muscato, and F. Russo, "Localization and self calibration of a robot for volcano exploration," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1. New Orleans, LA: IEEE Robotics and Automation, April 2004, pp. 586–591.

[11] DARPA, "DARPA Grand Challenge," http://www.darpa.mil/grandchallenge/index.asp, 2006.

[12] F. Dellaert, W. Burgard, D. Fox, and S. Thrun, "Using the Condensation Algorithm for Robust, Vision-based Mobile Robot Localization," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Ft. Collins, CO, June 1999, pp. 1–6.

[13] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm (with discussion)," *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

[14] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, November 2000.

[15] C. for Remote Sensing of Ice Sheets (CReSIS), "Center for Remote Sensing of Ice Sheets," URL: http://www.cresis.ku.edu, Lawrence, Kansas, 2007.

[16] D. Fox, W. Burgard, and S. Thrun, "Active Markov Localization for Mobile Robots," in *Robotics and Autonomous Systems*, vol. 25, 1998, pp. 195–207.

[17] D. Fox, W. Burgard, and S. Thrun, "Markov Localization for Mobile Robots in Dynamic Environments," *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999.

[18] D. Fox, J. Hightower, H. Kautz, L. Liao, and D. Patterson, "Bayesian Techniques for Location Estimation," in *Proceedings of The 2003 Workshop on Location-Aware Computing*, Seattle, WA, October 2003, pp. 16–18, part of the 2003 Ubiquitous Computing Conference.

[19] A. Giachetti, M. Campani, and V. Torre, "The use of optical flow for road navigation," *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 1, pp. 34–48, Feb 1998.

[20] C. M. Gifford and A. Agah, "Precise formation of multi-robot systems," in *Proceedings of the IEEE International Conference on Systems of Systems Engineering (SoSE)*, no. 105. San Antonio, Texas: Systems of Systems Engineering, April 2007, pp. 1–6.

[21] S. Gogineni, C. Allen, D. Braaten, T. Akins, P. Kanagaratnam, A. Agah, E. Akers, I. Bhattacharya, D.Dunson, V. Frost, H. Harmon, K. Jezek, K. Mason, A. Mohammad, R. Parthasarathy, S. Sivashanmugam, R. Stansbury, V. Ramasami, and C. Tsatsoulis, "Mobile Sensor Web for Polar Ice Sheet Measurements (ITR/SI+AP) PRISM 2004 Field Activities Report," University of Kansas, Lawrence, KS, Tech. Rep., November 2004.

[22] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. New Jersey: Prentice Hall, 2002.

[23] J.-S. Gutmann and D. Fox, "An Experimental Comparison of Localization Methods Continued," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Switzerland, October 2002, pp. 1–6.

[24] H. P. Harmon, R. S. Stansbury, E. L. Akers, and A. Agah, "Sensing and actuation for a polar mobile robot," in *Proceedings of the International Conference on Computing, Communications and Control Technologies (CCCT)*, vol. IV, Austin, Texas, August 2004, pp. 371–376.

[25] S. Hernandez, C. Morales, J. Torres, and L. Acosta, "A new localization system for autonomous robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2. Taiwan: IEEE Robotics and Automation, 2003, pp. 1588–1593.

[26] M.-K. Hu, "Visual pattern recognition by moment invariants," *Information Theory, IEEE Transactions on*, vol. 8, no. 2, pp. 178–187, Feb 1962.

[27] G. Jang, S. Lee, and I. Kweon, "Color landmark based self-localization for indoor mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1. Washington D.C.: IEEE Robotics and Automation, May 2002, pp. 1037–1042.

[28] P. Jensfelt, "Approaches to Mobile Robot Localization in Indoor Environments," Ph.D. dissertation, Signal, Sensors and Systems (S3), Royal Institute of Technology, SE-100 44 Stockholm, Sweden, 2001.

[29] M. Jogan, A. Leonardis, H. Wildenauer, and H. Bischof, "Mobile Robot Localization under Varying Illumination," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, Quebec, Canada, August 2002, pp. 1–4.

[30] J. Kosecka and F. Li, "Vision based topological Markov localization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2. IEEE Robotics and Automation, 2004, pp. 1481–1486.

[31] B. Kröse, N. Vlassis, R. Bunschoten, and Y. Motomura, "A probabilistic model for appearance-based robot localization," *Image and Vision Computing*, vol. 19, no. 6, pp. 381–391, April 2001. [Online]. Available: citeseer.ist.psu.edu/article/kr00probabilistic.html

[32] E. Krotkov, "Mobile robot localization using a single image," in *Proceedings. Robotics and Automation, IEEE International Conference on*, vol. 2. Arizona: IEEE Robotics and Automation, 1989, pp. 978–983.

[33] KU PRISM Team, "PRISM Home: Polar Radar for Ice Sheet Measurements," URL: http://www.ku-prism.org, Lawrence, Kansas, 2004.

[34] B. Kuipers and Y.-T. Byun, "A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations," *Journal of Robotics and autonomous Systems*, vol. 8, pp. 47–63, 1991.

[35] A. Lankenau and T. Rofer, "A versatile and safe mobility assistant," in *Robotics and Automation Magazine, IEEE*, March 2001, vol. 8, no. 1, pp. 29–37.

[36] A. Lankenau and T. Rofer, "Mobile robot self-localization in large-scale environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2. Washington D.C.: IEEE Robotics and Automation, May 2002, pp. 1359–1364.

[37] D. Lee, W. Chung, and M. Kim, "A reliable position estimation method of the service robot by map matching," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2. IEEE Robotics and Automation, 2003, pp. 2830–2835.

[38] A. Leonardis and H. Bischof, "Robust recognition using eigenimages," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 99–118, 2000.

[39] K. Lingemann, H. Surmann, A. Nuchter, and J. Hertzberg, "Indoor and outdoor localization for fast mobile robots," in *Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3, no. 28. IEEE Intelligent Robots and Systems, 2004, pp. 2185–2190.

[40] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 1999, pp. 1150–1157. [Online]. Available: citeseer.ist.psu.edu/lowe99object.html

[41] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[42] D. Lowe, URL: http://www.cs.ubc.ca/~lowe, University of British Columbia, 2007.

[43] R. Madhavan, K. Fregene, and L. Parker, "Distributed heterogeneous outdoor multi-robot localization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1. Washington D.C.: IEEE Robotics and Automation, May 2002, pp. 374–381.

[44] J. Meltzer, R. Gupta, M.-H. Yang, and S. Soatto, "Simultaneous localization and mapping using multiple view feature descriptors," in *Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 2. Japan: IEEE Intelligent Robots and Systems, October 2004, pp. 1550–1555.

[45] Q.-H. Meng, Y.-C. Sun, and Z.-L. Cao, "Adaptive extended Kalman filter (AEKF)-based mobile robot localization using sonar," *Robotica*, vol. 18, no. 5, pp. 459–473, 2000.

[46] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[47] M. Montemerlo and S. Thrun, "A Multi-Resolution Pyramid for Outdoor Robot Terrain Perception," in *Proceedings of the AAAI National Conference on Artificial Intelligence*. San Jose, CA: AAAI, July 2004, pp. 1–6.

[48] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-D objects from appearance," *International Journal of Computer Vision*, vol. 14, no. 1, pp. 5–24, 1995.

[49] J. Neira, J. Tardos, J. Horn, and G. Schmidt, "Fusing range and intensity images for mobile robot localization," *Robotics and Automation, IEEE Transactions on*, vol. 15, no. 1, pp. 76–84, Feb 1999.

[50] I. Nourbakhsh, R. Powers, and S. Birchfield, "DERVISH an Office-Navigating Robot," *AI Magazine*, vol. 16, pp. 53–60, 1995.

[51] OpenCV, "OpenCV: Open Computer Vision Library," http://sourceforge.net/projects/opencvlibrary.

[52] J. Porta, J. Verbeek, and B. Krse, "Active Appearance-Based Robot Localization Using Stereo Vision," *Autonomous Robots*, vol. 18, no. 1, pp. 59–80, 2005.

[53] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[54] A. Robotics, URL: http://www.activrobots.com, Amherst, New Hampshire, 2007.

[55] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, New Jersey: Pearson Education, 2003.

[56] R. Siegwart and I. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, ser. Intelligent Robotics and Autonomous Agents series. Cambridge, Massachusetts: MIT Press, 2004.

[57] R. Sim and G. Dudek, "Comparing image-based localization methods," in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, August 2003, pp. 1560–1562.

[58] R. Simmons and S. Koenig, "Probabilistic Robot Navigation in Partially Observable Environments," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Palo Alto, CA, 1995, pp. 1080–1087. [Online]. Available: citeseer.ist.psu.edu/simmons95probabilistic.html

[59] R. S. Stansbury, E. L. Akers, H. P. Harmon, and A. Agah, "Survivability, Mobility, and Functionality of a rover for radars in polar regions," *International Journal of Control, Automation, and Systems*, vol. 2, no. 3, pp. 343–353, September 2004.

[60] R. Stansbury, E. L. Akers, H. P. Harmon, and A. Agah, "Simulation and testbeds of autonomous robots in harsh environments," in *Software Engineering for Experimental Robotics*, ser. STAR, D. Brugali, Ed. Springer-Verlag Berlin Heidelberg, 2007, vol. 30, pp. 71–92.

[61] D. Strelow and S. Singh, "Optimal motion estimation from visual and inertial measurements," in *Proceedings of the Workshop on Applications of Computer Vision*, Orlando, FL, December 3-4 2002, pp. 314–319.

[62] R. Thrapp, C. Westbrook, and D. Subramanian, "Robust localization algorithms for an autonomous campus tour guide," in *Proceedings of the*

IEEE International Conference on Robotics and Automation (ICRA), vol. 2. Seoul, Korea: IEEE Robotics and Automation, May 21-26 2001, pp. 2065–2071.

[63] S. Thrun, "Probabilistic Algorithms in Robotics," *AI Magazine*, vol. 21, no. 4, pp. 93–109, 2000.

[64] S. Thrun, "Robotic Mapping: A Survey," in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002.

[65] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "MINERVA: A second generation mobile tour-guide robot," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[66] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo Localization for Mobile Robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2000.

[67] S. Thrun, J.-S. Gutmann, D. Fox, W. Burgard, and B. Kuipers, "Integrating topological and metric maps for mobile robot navigation: A statistical approach," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Orlando, FL, October 1998, pp. 989–996.

[68] A. Torralba, K. Murphy, W. Freeman, and M. Rubin, "Context-based vision system for place and object recognition," in *International Conference on Computer Vision*, Nice, France, October 2003. [Online]. Available: citeseer.ist.psu.edu/torralba03contextbased.html

[69] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2. San Fransisco, CA: IEEE Robotics and Automation, April 2000, pp. 1023–1029.

[70] J. Wolf, W. Burgard, and H. Burkhardt, "Robust vision-based localization for mobile robots using an image retrieval system based on invariant features," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1. Washington D.C.: IEEE Robotics and Automation, May 2002, pp. 359–365.

Appendix A

Topological Test Results

Feature	Class	Total Images	Number Correct	Percent Localized	Best Case	Worst Case
Gray 1x1	HMM	2480	2162	0.87177	230	200
	ML	2480	1761	0.71008	190	165
	Bayes	2480	1760	0.70968	190	165
Color 1x1	HMM	2290	2176	0.95022	221	213
	ML	2290	2017	0.88079	208	196
	Bayes	2290	2015	0.87991	208	196
Gray 2x2	HMM	2270	2109	0.92907	214	207
	ML	2270	1922	0.8467	200	179
	Bayes	2270	1916	0.84405	199	179
Color 2x2	HMM	2300	2159	0.9387	221	208
	ML	2300	1980	0.86087	207	189
	Bayes	2300	1975	0.8587	205	188
Gray 3x3	HMM	2660	2477	0.9312	256	242
	ML	2660	2293	0.86203	236	220
	Bayes	2660	2289	0.86053	236	220
Color 3x3	HMM	2410	2295	0.95228	232	225
	ML	2410	2103	0.87261	220	203
	Bayes	2410	2092	0.86805	220	201
Hu Moments Gray	HMM	2150	434	0.20186	50	38
	ML	2150	192	0.0893	25	13
	Bayes	2150	192	0.0893	25	13
Hu Moments RGB	HMM	2190	1305	0.59589	154	112
	ML	2190	600	0.27397	72	51
	Bayes	2190	600	0.27397	72	51
Eigenimages	HMM	2230	1817	0.8148	193	170
	ML	2230	1520	0.68161	161	142
	Bayes	2230	1520	0.68161	161	142

Table A.1: Topological test one results: localization accuracy

Feature	Class	Total Images	Time To Localize		
			Ave	Best	Worst
Gray 1x1	HMM	2480	1.3	1	2
	ML	2480	1.2	1	2
	Bayes	2480	1.2	1	2
Color 1x1	HMM	2290	1.2	1	2
	ML	2290	1.1	1	2
	Bayes	2290	1.1	1	2
Gray 2x2	HMM	2270	1.4	1	3
	ML	2270	1.1	1	2
	Bayes	2270	1.1	1	2
Color 2x2	HMM	2300	1.4	1	3
	ML	2300	1.2	1	2
	Bayes	2300	1.2	1	2
Gray 3x3	HMM	2660	1.9	1	4
	ML	2660	1.0	1	1
	Bayes	2660	1.0	1	1
Color 3x3	HMM	2410	1.5	1	2
	ML	2410	1.2	1	2
	Bayes	2410	1.2	1	2
Hu Moments Gray	HMM	2150	21.0	8	39
	ML	2150	24.1	1	51
	Bayes	2150	24.1	1	51
Hu Moments RGB	HMM	2190	4.8	2	15
	ML	2190	10.1	3	28
	Bayes	2190	10.1	3	28
Eigenimages	HMM	2230	1.3	1	2
	ML	2230	1.3	1	2
	Bayes	2230	1.3	1	2

Table A.2: Topological test one results: global localization time

Feature	Class	Total Images	Number Correct	Percent Localized	Best Case	Worst Case
Gray 1x1	HMM	3930	3572	0.90891	367	344
	ML	3930	2915	0.74173	306	269
	Bayes	3930	2910	0.74046	306	269
Color 1x1	HMM	4370	4162	0.9524	424	410
	ML	4370	3755	0.85927	382	369
	Bayes	4370	3750	0.85812	382	368
Gray 2x2	HMM	3860	3623	0.9386	371	355
	ML	3860	3242	0.8399	331	308
	Bayes	3860	3237	0.8386	330	308
Color 2x2	HMM	4080	3863	0.94681	393	377
	ML	4080	3535	0.86642	370	342
	Bayes	4080	3526	0.86422	370	340
Gray 3x3	HMM	3860	3651	0.94585	376	353
	ML	3860	3185	0.82513	337	299
	Bayes	3860	3182	0.82435	337	299
Color 3x3	HMM	3710	3485	0.93935	358	336
	ML	3710	3150	0.84906	334	292
	Bayes	3710	3141	0.84663	333	290
Hu Moments Gray	HMM	4190	591	0.14105	69	45
	ML	4190	376	0.08974	49	28
	Bayes	4190	376	0.08974	49	28
Hu Moments RGB	HMM	4070	2521	0.61941	302	218
	ML	4070	1494	0.36708	161	136
	Bayes	4070	1494	0.36708	161	136
Eigenimages	HMM	3810	3228	0.84724	339	308
	ML	3810	2738	0.71864	286	265
	Bayes	3810	2737	0.71837	286	265

Table A.3: Topological test two results: localization accuracy

Feature	Class	Total Images	Time To Localize		
			Ave	Best	Worst
Gray 1x1	HMM	3930	2.2	1	9
	ML	3930	1.3	1	3
	Bayes	3930	1.3	1	3
Color 1x1	HMM	4370	1.4	1	2
	ML	4370	1.2	1	2
	Bayes	4370	1.2	1	2
Gray 2x2	HMM	3860	1.6	1	3
	ML	3860	1.2	1	2
	Bayes	3860	1.2	1	2
Color 2x2	HMM	4080	1.4	1	3
	ML	4080	1.1	1	2
	Bayes	4080	1.1	1	2
Gray 3x3	HMM	3860	1.6	1	4
	ML	3860	1.4	1	3
	Bayes	3860	1.4	1	3
Color 3x3	HMM	3710	1.6	1	3
	ML	3710	1.4	1	3
	Bayes	3710	1.4	1	3
Hu Moments Gray	HMM	4190	17.0	7	31
	ML	4190	20.0	3	56
	Bayes	4190	20.0	3	56
Hu Moments RGB	HMM	4070	4.9	1	8
	ML	4070	10.1	1	27
	Bayes	4070	10.1	1	27
Eigenimages	HMM	3810	1.4	1	3
	ML	3810	1.2	1	2
	Bayes	3810	1.2	1	2

Table A.4: Topological test two results: global localization time

Feature	Class	Total Images	Number Correct	Percent Localized	Best Case	Worst Case
Gray 1x1	HMM	7180	6450	0.89833	659	627
	ML	7180	5152	0.71755	546	473
	Bayes	7180	5148	0.71699	546	473
Color 1x1	HMM	7190	6830	0.94993	694	661
	ML	7190	6315	0.8783	644	615
	Bayes	7190	6302	0.8765	644	615
Gray 2x2	HMM	7260	6875	0.94697	693	679
	ML	7260	6192	0.85289	632	593
	Bayes	7260	6185	0.85193	631	593
Color 2x2	HMM	7130	6748	0.94642	684	665
	ML	7130	6224	0.87293	651	598
	Bayes	7130	6214	0.87153	648	598
Gray 3x3	HMM	6990	6554	0.93763	673	640
	ML	6990	6044	0.86466	615	591
	Bayes	6990	6038	0.86381	616	591
Color 3x3	HMM	7490	7025	0.93792	717	684
	ML	7490	6519	0.87036	673	621
	Bayes	7490	6505	0.86849	671	619
Hu Moments Gray	HMM	8020	1578	0.19676	193	134
	ML	8020	978	0.12195	110	86
	Bayes	8020	978	0.12195	110	86
Hu Moments RGB	HMM	7330	4606	0.62838	506	411
	ML	7330	2083	0.28417	222	187
	Bayes	7330	2083	0.28417	222	187
Eigenimages	HMM	7080	6010	0.84887	617	585
	ML	7080	4949	0.69901	522	461
	Bayes	7080	4949	0.69901	522	461

Table A.5: Topological test three results: localization accuracy

Feature	Class	Total Images	Time To Localize		
			Ave	Best	Worst
Gray 1x1	HMM	7180	3.4	1	15
	ML	7180	1.5	1	3
	Bayes	7180	1.5	1	3
Color 1x1	HMM	7190	1.1	1	2
	ML	7190	1.0	1	1
	Bayes	7190	1.0	1	1
Gray 2x2	HMM	7260	1.4	1	3
	ML	7260	1.3	1	2
	Bayes	7260	1.3	1	2
Color 2x2	HMM	7130	1.6	1	4
	ML	7130	1.4	1	3
	Bayes	7130	1.4	1	3
Gray 3x3	HMM	6990	2.0	1	5
	ML	6990	1.6	1	3
	Bayes	6990	1.6	1	3
Color 3x3	HMM	7490	2.3	1	6
	ML	7490	1.3	1	2
	Bayes	7490	1.3	1	2
Hu Moments Gray	HMM	8020	90.2	56	125
	ML	8020	4.7	1	10
	Bayes	8020	4.7	1	10
Hu Moments RGB	HMM	7330	23.7	2	32
	ML	7330	2.5	1	5
	Bayes	7330	2.5	1	5
Eigenimages	HMM	7080	5.5	1	12
	ML	7080	3.4	1	10
	Bayes	7080	3.4	1	10

Table A.6: Topological test three results: global localization time

Feature	Classifier	Kidnapped Robot Problem		
		Average	**Best**	**Worst**
Gray 1x1	HMM	3.7	2	8
	ML	1.3	1	2
	Bayes	1.3	1	2
Color 1x1	HMM	2.3	2	3
	ML	1.4	1	3
	Bayes	1.4	1	3
Gray 2x2	HMM	3	2	7
	ML	1.1	1	2
	Bayes	1.1	1	2
Color 2x2	HMM	2.3	2	3
	ML	1.1	1	2
	Bayes	1.1	1	2
Gray 3x3	HMM	2.6	2	4
	ML	1.1	1	2
	Bayes	1.1	1	2
Color 3x3	HMM	2.5	2	3
	ML	1.2	1	2
	Bayes	1.3	1	2
Hu Moments Gray	HMM	76.3	16	Never (116)
	ML	35.1	2	Never (116)
	Bayes	35.1	2	Never (116)
Hu Moments RGB	HMM	28.1	8	46
	ML	12.3	1	30
	Bayes	12.3	1	30
Eigenimages	HMM	2.2	2	3
	ML	1.2	1	3
	Bayes	1.2	1	3

Table A.7: Topological test four results: kidnapped robot problem

Feature	Classifier	Kidnapped Robot Problem		
		Average	**Best**	**Worst**
Gray 1x1	HMM	2.7	2	4
	ML	1.2	1	3
	MAP	1.2	1	3
Color 1x1	HMM	3.4	2	6
	ML	1.3	1	2
	MAP	1.3	1	2
Gray 2x2	HMM	3.2	2	5
	ML	1.5	1	3
	MAP	1.5	1	3
Color 2x2	HMM	2.8	2	5
	ML	1.1	1	2
	MAP	1.1	1	2
Gray 3x3	HMM	3.3	2	7
	ML	1.1	1	2
	MAP	1.1	1	2
Color 3x3	HMM	3.2	2	7
	ML	1	1	1
	MAP	1	1	1
Hu Moments Gray	HMM	179.7	86	210
	ML	87	68	110
	MAP	87	68	110
Hu Moments RGB	HMM	13.9	4	41
	ML	1.6	1	3
	MAP	1.6	1	3
Eigenimages	HMM	2.8	2	5
	ML	1.2	1	2
	MAP	1.2	1	2

Table A.8: Topological test five results: kidnapped robot problem

Feature	Class	Total Images	Number Correct	Percent Localized	Best Case	Worst Case
Color 1x1	HMM	8410	8083	0.9611	819	798
	ML	8410	7728	0.9189	768	760
Gray 2x2	HMM	8740	8386	0.9595	846	828
	ML	8740	7963	0.9111	810	768
Color 2x2	HMM	8510	8212	0.9650	832	807
	ML	8510	7697	0.9045	787	760
Gray 3x3	HMM	8860	8462	0.9551	854	839
	ML	8860	7938	0.8959	811	782

Table A.9: Topological test six results: localization accuracy

Feature	Class	Total Images	Time To Localize		
			Ave	Best	Worst
Color 1x1	HMM	8410	1.4	1	3
	ML	8410	1.1	1	2
Gray 2x2	HMM	8740	2.1	1	5
	ML	8740	1.3	1	3
Color 2x2	HMM	8510	1.7	1	4
	ML	8510	1.2	1	2
Gray 3x3	HMM	8860	1.7	1	4
	ML	8860	1.4	1	2

Table A.10: Topological test six results: global localization time

Feature	Class	Total Images	Number Correct	Percent Localized	Best Case	Worst Case
Color 1x1	HMM	2200	2130	0.9682	217	210
	ML	2200	1951	0.8868	204	185
Gray 2x2	HMM	1890	1793	0.9487	184	173
	ML	1890	1687	0.8926	176	159
Color 2x2	HMM	1800	1739	0.9661	178	167
	ML	1800	1608	0.8933	168	149
Gray 3x3	HMM	2190	2088	0.9534	214	198
	ML	2190	1924	0.8785	207	175

Table A.11: Topological test seven results: localization accuracy

Feature	Class	Total Images	Time To Localize		
			Ave	Best	Worst
Color 1x1	HMM	8410	1.4	1	3
	ML	8410	1.1	1	2
Gray 2x2	HMM	8740	1.3	1	2
	ML	8740	1.3	1	2
Color 2x2	HMM	8510	1.2	1	2
	ML	8510	1.1	1	2
Gray 3x3	HMM	8860	1.1	1	2
	ML	8860	1.1	1	2

Table A.12: Topological test seven results: global localization time

Feature	Class	Total Images	Number Correct	Percent Localized	Best Case	Worst Case
Eigenimages	HMM	1290	334	0.2589	53	25
	ML	1290	438	0.3395	58	36
Color 1x1	HMM	1270	311	0.2449	48	2
	ML	1270	321	0.2528	40	24
Gray 1x1	HMM	1470	324	0.2204	67	20
	ML	1470	316	0.2150	47	20

Table A.13: Topological polar results: localization accuracy

Feature	Class	Total Images	Time To Localize		
			Ave	Best	Worst
Eigenimages	HMM	1290	63.9	30	86
	ML	1290	1.6	1	4
Color 1x1	HMM	1270	42.8	25	53
	ML	1270	1.0	1	1
Gray 1x1	HMM	1470	24.3	21	49
	ML	1470	11.0	1	21

Table A.14: Topological polar results: localization time

Appendix B

Geometric Test Results

TEST	Total Images	Correct Location	Correct Orientation	Accuracy	
				Location	Orientation
Indoor 3ft Grid Spacing	3360	3263	3109	0.9711	0.9253
Indoor 3ft Grid Spacing with Noise	3357	3066	2269	0.9133	0.6759
Indoor 1ft Grid Spacing	3360	3170	3037	0.9435	0.9039
Indoor 1ft Grid Spacing with Noise	3360	3053	2377	0.9086	0.7074
Outdoor 1ft Grid Spacing	3353	3298	3325	0.9836	0.9916
Outdoor 1ft Grid Spacing with Noise	3353	3345	3350	0.9976	0.9991

Table B.1: Geometric test results: position and orientation accuracy