Topological Localization Using Appearance-Based Recognition

Received 20th July 2009; accepted 3rd December 2009.

Eric L. Akers, Arvin Agah

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.

Sophisticated localization systems can solve both the global location problem and the kid-napped 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.

CReSIS (Center for Remote Sensing of Ice Sheets) 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 need to be able to return to campsites 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 dfferent locations of the campsites is the beginning of this process.

The approach is to use a single camera for use in multiple types of large-scale environments, indoors and outdoors using a topological map. The system described 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. Using these simple features, the system is able to determine its location within the map at 95% accuracy.

Keywords: *topological localization, mobile robotics, computer vision, appearance-based methods.*

1. Introduction

The localization of mobile robots is the fundamental problem of determining the position and orientation of the robot. Researchers have looked into many different methods of solving the problem several kinds of sensors.

The first solutions to this problem relied on techniques that had been used in the past by sailors and the military. These solutions are sometimes referred to as position tracking. These require that you begin in a known location; and by measuring how far you have gone in a specific direction, you can determine where you are.

These solutions usually rely on odometry or inertial measuring units. Odometry is the measuring of the rotation of the wheels to measure how far the robot has gone. These systems usually accumulate error as they move because of the slippage of wheels, especially when turning. Inertial measuring units tend to drift after some time, also causing errors.

Most solutions usually rely on a specific type of environment such as indoor environments. Few solutions have worked both indoors and outdoors.

More recently, SLAM (simultaneous localization and mapping) has become more popular. This is a problem where the robot is placed into a new environment without a map, and it must be able to both map an area and localize itself at the same time. This could be a requirement of a search and rescue robot. Some claim that in order to have a truly autonomous robot, the robot must be able to solve this problem.

However, many robots will always require a priori maps in order to be able to perform a specific task such as a delivery robot or the autonomous vehicles in the Darpa Grand Challenge [10].

1.1. Topological Localization

Many localization systems can be described by the type of map that is being used: geometric, topological, or a hybrid map. This work uses a topological approach with the intention of using a hybrid approach later on. A topological map is just an adjacency graph. Nodes are connected to other nodes with edges. A pure topological map contains no information about size or the distance between nodes. However, as part of a navigation system, some extra information such how to move from one location to another may be stored inside the map as well. Some examples of a topological localization system include [31], [17], [29], [5].

In contrast, a geometric based system relies on maps that show edges, lines, and geometric objects that represent the map. These maps contain more information and can localize to a much more fine position. The topological system can tell you what location you are in, but not necessarily where in that location. Some geometric systems are described in [7], [28], [1].

The goals of these systems are somewhat different. The goal of a topological system is to tell you which location you are in, such as room 314 in Nichols Hall. The goal of a geometric system is to give a precise determination of the current location within a specific area.

The hybrid approach attempts to utilize both methods. In this approach, the topological map usually stores a geometric map of its area.

1.2. Motivation

This paper describes the research working towards a localization system that can operate in several di erent environments using only a camera. This kind of a system has the advantages of being able to use a simple and relatively inexpensive sensor that is easily portable. It can be used standalone or in addition to another system to improve the reliability.

1.2.1 PRISM/CReSIS

The Polar Radar for Ice Sheet Measurements (PRISM) project is [19] is part of the Center for Remote Sensing of Ice Sheets (CReSIS) [9] at the University of Kansas. This project's goal is to develop radar systems to measure polar ice sheet properties in order to accurately determine their mass and other characteristics. Such 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 [19]. In order to accommodate these radar systems, an autonomous robot 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 GPS for navigation. However, once the robot returns to camp, it is desirable for the vehicle to use other sensors for navigation in order to drive through the camp safely and accurately. A system like the one described in this paper will allow the robot to safely find the fuel station and the location for unloading the data without driving through tent city, the area used for gathering snow 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 collecting data again can be greatly reduced and require less
human assistance. In order to operate autonomously for several weeks at a time, this is a task that must be solved by the robot [3].

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.

1.2.2. Service Robots

Service robots provide assistance to humans in many different ways. Wheel chair service robots have been used such as the Bremen Autonomous Wheelchair as discussed in [20].

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 told 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 e ort.

Uninhabited Ground Vehicles (UGV)

UGVs have been tested recently in the DARPA Grand Challenge [10] 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 effciency.

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.

2. Related Work

Related works are described in this section to provide a brief background of some of the projects that have done research in the area of localization and mapping.

2.1. 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 [23].

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 [7]. 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 on-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 [8] 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 struc-ture 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 [28]. 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.

2.2. Appearance-Based Localization

At the University of Amsterdam an active appearance-based method has been used for localization of a robot named Lino [24]. 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 land-mark 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 proba-bilistic 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 [16]. 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 [21] 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 [31]. 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, lumi-nance, 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.

3. 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 campsites in Greenland and Antarctica. A topological map is used where each node in the map is represented by a set of images. The problem then becomes to recognize specific locations within the topological map based in images captured from the camera on the robot. A probabilistic localization system was designed that uses appearance-based features and a hidden Markov model as the classifier to attempt to solve this problem.

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 set of Gaussian mixture models. 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 evaluation.

3.1. 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 [31], [24], [18]. 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 [14], Hu Moments [14], and eigenimages [21, [22].

Using pixel values in images that 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. [15] 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 adescriptor that is much smaller than the image itself.

3.1.1. 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 [11], [26] algorithm is used to generate the parameters of the mixture, which include the mean, covariance, and weight. The program described in [6] 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 1.

$$P\,(X = x|Q = q) = \sum_k w_k * N_k \tag{1}$$

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.

3.2. 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 [25]. The HMM relies on the Markov assumption, that the value of the next state is dependent only on the value of the current state [26].

The hidden Markov model-decoding algorithm described in [12] on page 135 was used for determining which node in the map has the highest probability. This decoding problem is that 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 o the hidden Markov model and acting more like the naive Bayes classifier.

Table 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.

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

The ML classifier selects the node with the highest likelihood (from the GMM) and has no memory, see Equation 2. In contrast, 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 3.

$$\text{ML Location} = argmax[P\,(X = x\,|\,Q = q)] \tag{2}$$

$$\text{Naive Bayes Location} = argmax[áP\;(X = x\,|\,Q = q) * P\;(Q)] \tag{3}$$

3.2.1. 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 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 mode 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 ifthe current location has many connections. This would mean that each of these connections is 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 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).

3.3. Topological Maps

Two different maps were used to test the system. The first map, see Figure 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

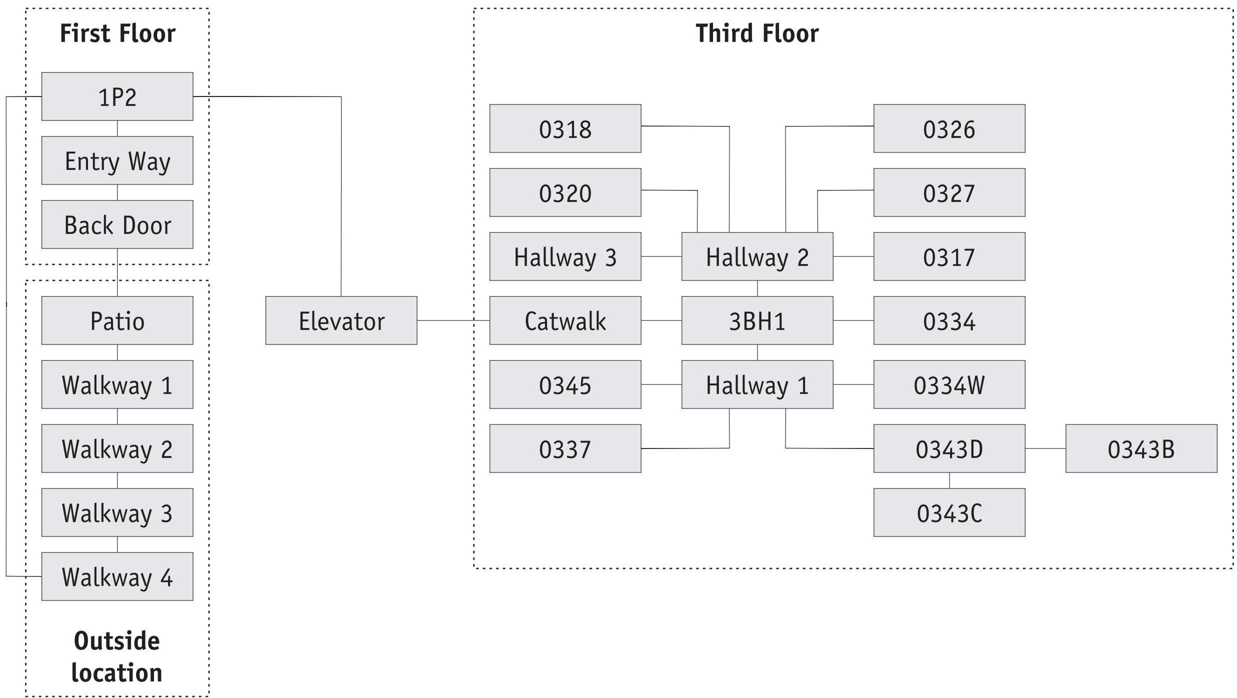

Fig. 1. Adjacency graph representing the first topological map used for evaluation. The building has three floors, but the map was created from locations on only the first and third floors.

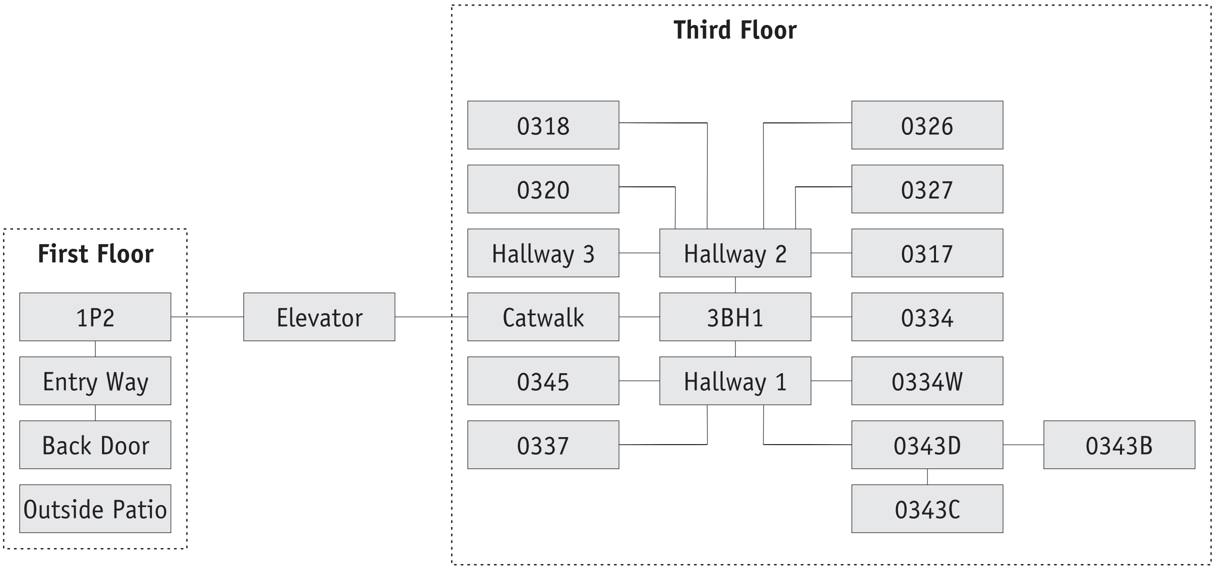

Fig. 2. Adjacency graph representing the second topological map used for evaluation. This map is broken into three different general locations: first floor, third floor, and the outside areas.

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 3, 4, 5, and 6 show a sample of some of the images used to make up inside nodes 334, 335, Hallway 1, and Hallway 2, respectively. 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 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 1, does not give a good indication of how well the system works outdoors. Figures 7 to 9 show images from some of the added outside locations.

3.4. Features

Several feature descriptors were used to test the system, color and gray histograms [14], Hu Moments [15], and eigenimages [21], [22]. Variants of the color and gray histograms were also used. Table 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 his-

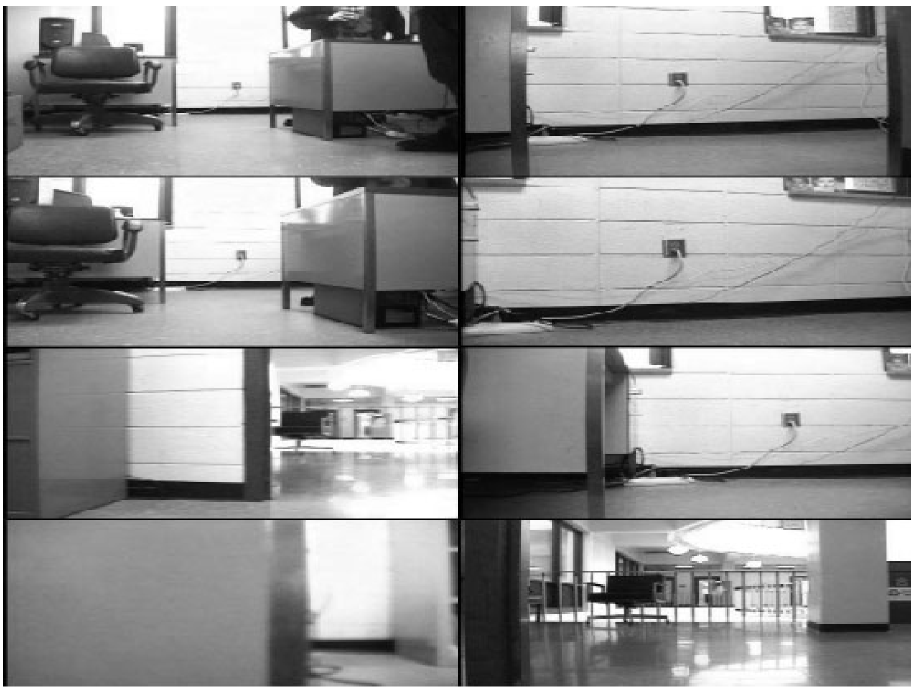

Fig. 3. Room 334 sample images.

Fig. 4. Room 335 sample images.

Fig. 5. Hallway 1 sample images.

togram. 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 bin 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 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 calculated each of the seven Hu Moments on each band of the RGB image separately, and appended them similar to the color histograms. Therefore, the color Hu

Fig. 6. Hallway 2 sample images.

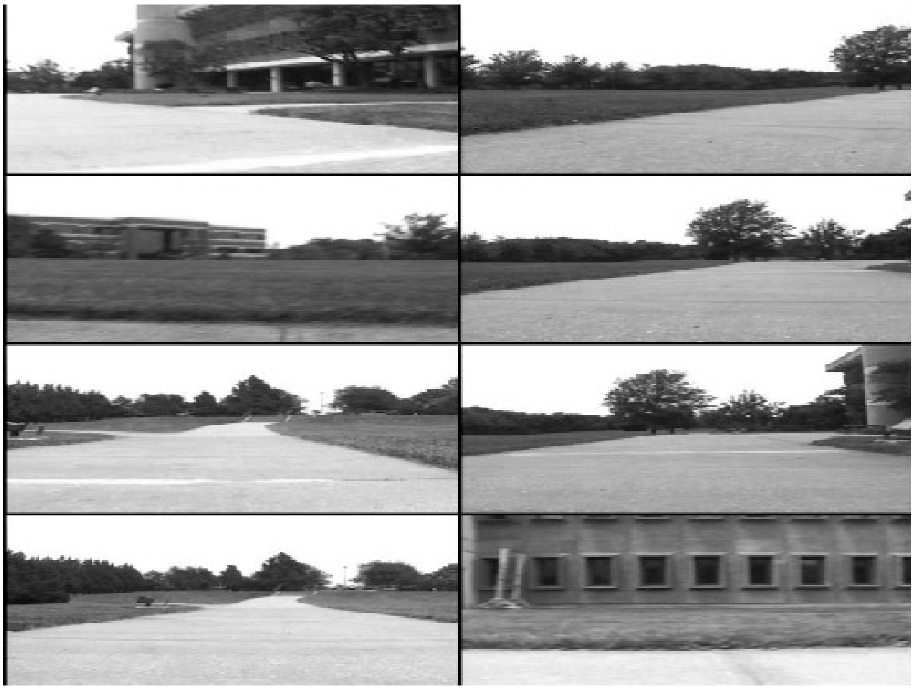

Fig. 7. Walkway 1 sample images.

Moments descriptor was three times the size of the gray Hu Moments descriptor.

Table 2. Variations of the color and gray histogram feature descriptors.

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

The eigenimage descriptor is described in [22]. 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.

3.4.1. Principle Component Analysis

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 [27] that 20 components were sufficient to describe the eigenimage descriptors. The color and gray histo-

Fig. 8. Walkway 2 sample images.

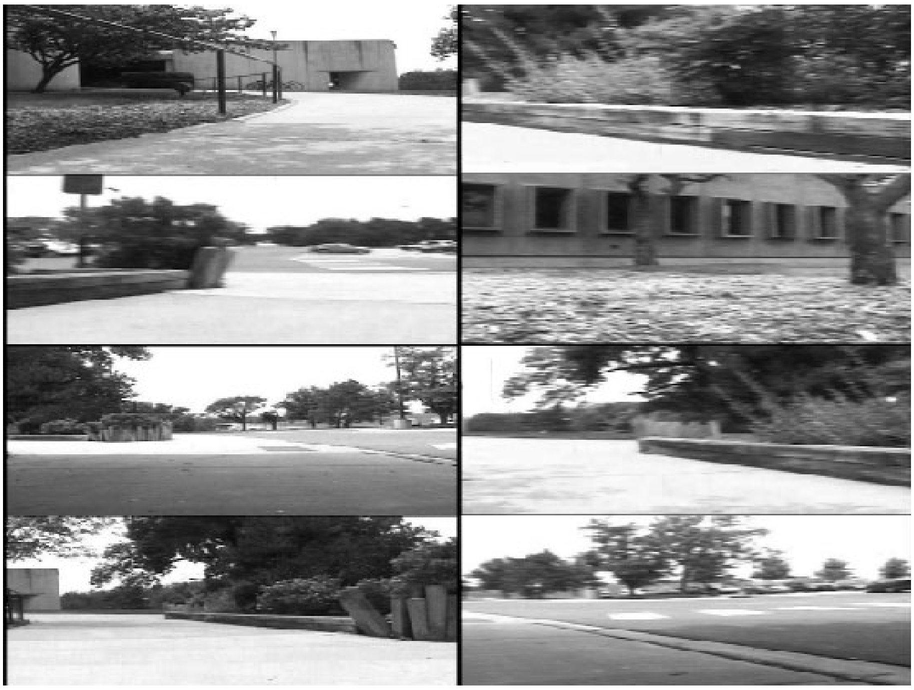

Fig. 9. Walkway 3 sample images.

grams 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 [30].

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 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.

Table 3. Feature descriptor descriptions. The size of the descriptors before and after PCA has been performed.

	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

4. 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 was 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 o ine. 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 3.2.

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.

Table 4. Nodes visited for each test run in the order they were visited. Four and five were used for the kidnapped robot tests. The asterisk represents the node in which the robot was 'teleported' to another location not directly adjacent to its previous. Six and seven were used for the outdoor tests.

Test	Nodes Visited (in order)
One	0343C, 0343D, 0343B, 0343D, HALLWAY 1, 3BH1, HALLWAY 2, HALLWAY 3, HALLWAY 2, 0320
Two	0343C, 0343D, 0343B, 0343D, HALLWAY 1, 0337, HALLWAY 1, 3BH1, 0334, 3BH1, CATWALK3, ELEVATOR, 1P2, ENTRY WAY, BACK DOOR, PATIO, BACK DOOR, ENTRYWAY
Three	HALLWAY 3, HALLWAY 2, 0326, HALLWAY 2, 0317, HALLWAY 2, 0318, HALLWAY 2, 0320, HALLWAY 2, 0327, HALLWAY 2, 3BH1, HALLWAY 1, 0337, HALLWAY 1, 0344W, HALLWAY 1, 0345, HALLWAY 1, 0343D, 0343C, 0343D, 0343B,0343D, HALLWAY 1, 3BH1, CATWALK3, ELEVATOR, 1P2, ENTRY WAY, BACK DOOR, PATIO
Four	0343C, 0343D, 0343B, 0343D, HALLWAY 1, 1P2*, ENTRY, BACK DOOR, PATIO, BACK DOOR
Five	0327, HALLWAY 2, 0317, HALLWAY 2, 0327, 0327, BACK DOOR*, PATIO, BASK DOOR, ENTRY, 1P2, ELEVATOR, CATWALK3, 3BH1, HALLWAY 2, 0327
Six	HALLWAY 3, HALLWAY 2, 0326, HALLWAY 2, 0317, HALLWAY 2, 0318, HALLWAY 2, 0320, HALLWAY 2, 0327, HALLWAY 2, 3BH1, HALLWAY 1, 0337, HALLWAY 1, 0344W, HALLWAY 1, 0345, HALLWAY 1, 0343D, 0343C, 0343D, 0343B, 0343D, HALLWAY 1, 3BH1, CATWALK3, ELEVATOR, 1P2, ENTRY WAY, BACK DOOR, PATIO, WALKWAY 1, WALKWAY 2, WALKWAY 3, WALKWAY 4
Seven	PATIO, WALKWAY 1, WALKWAY 2, WALKWAY 3, WALKWAY 4, WALKWAY 3, WALKWAY 2, WALKWAY 1, PATIO

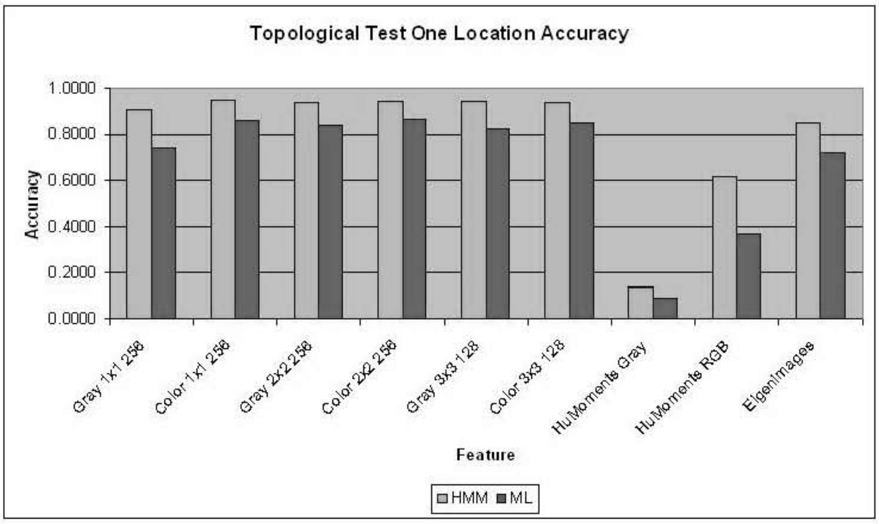

Fig. 10. Topological test one: position accuracy.

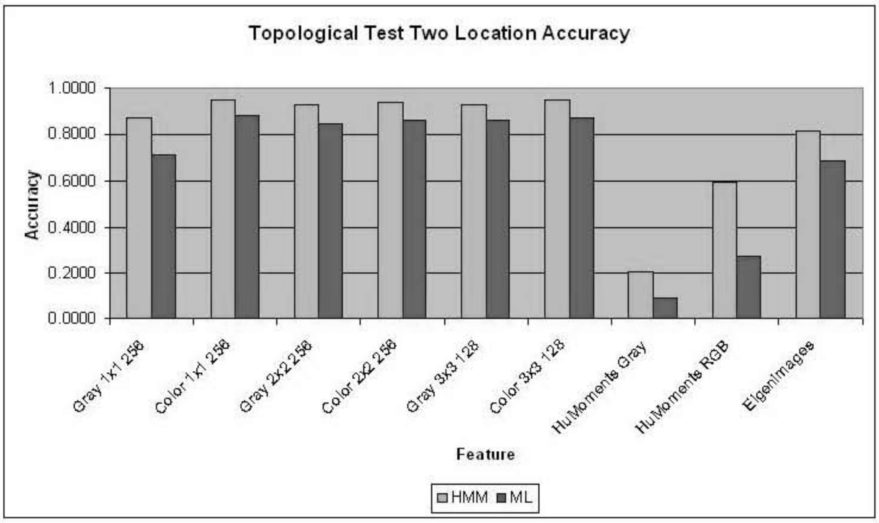

Fig. 11. Topological test two: position accuracy.

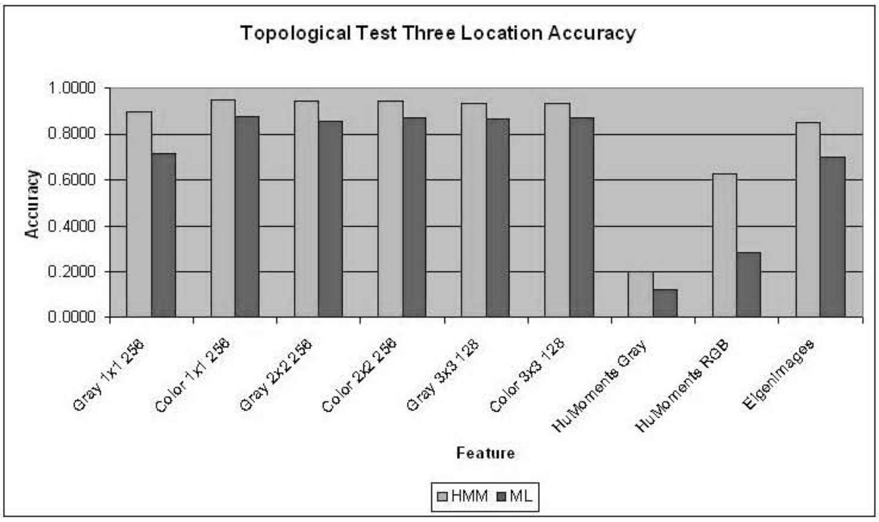

Fig. 12. Topological test three: position accuracy.

4.1. Testing Results

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 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 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 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 loca-tion 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 1 includes a reference of the locations visited.

Figures 10 to 12 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 outperforms 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 13 to 15 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.

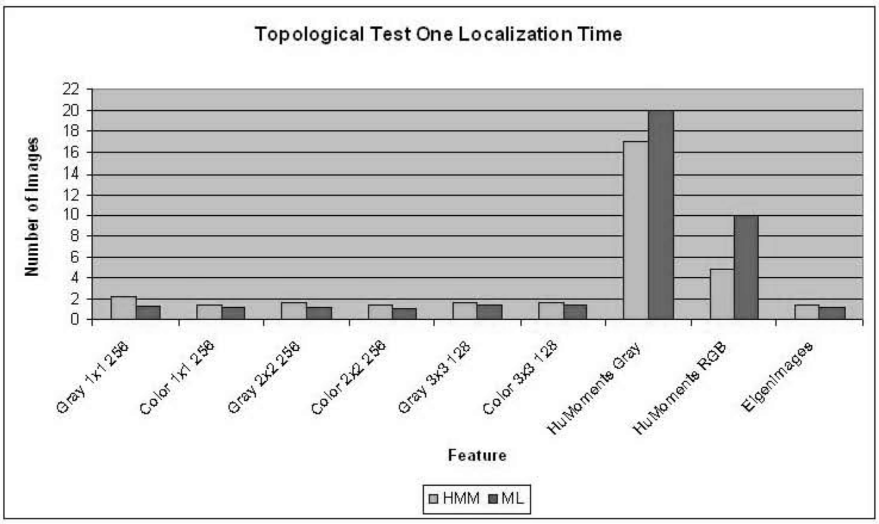

Fig. 13. Topological test one: localization time.

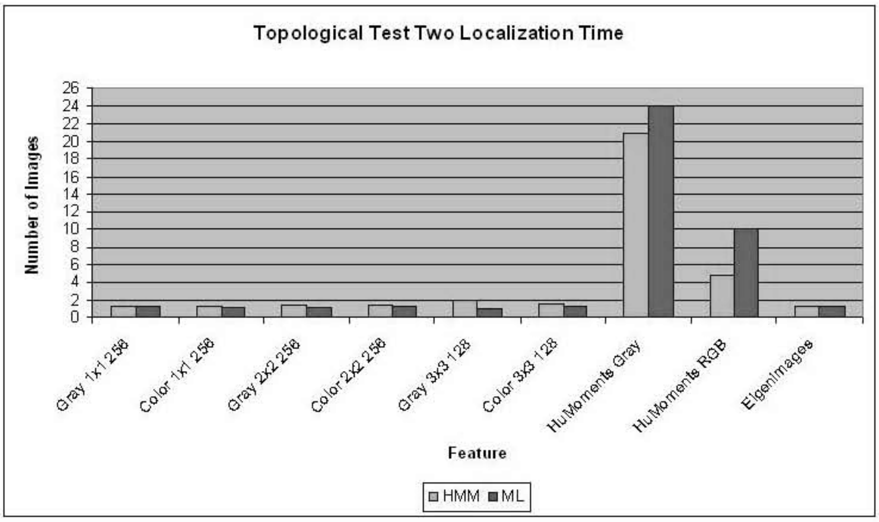

Fig. 14. Topological test two: localization time.

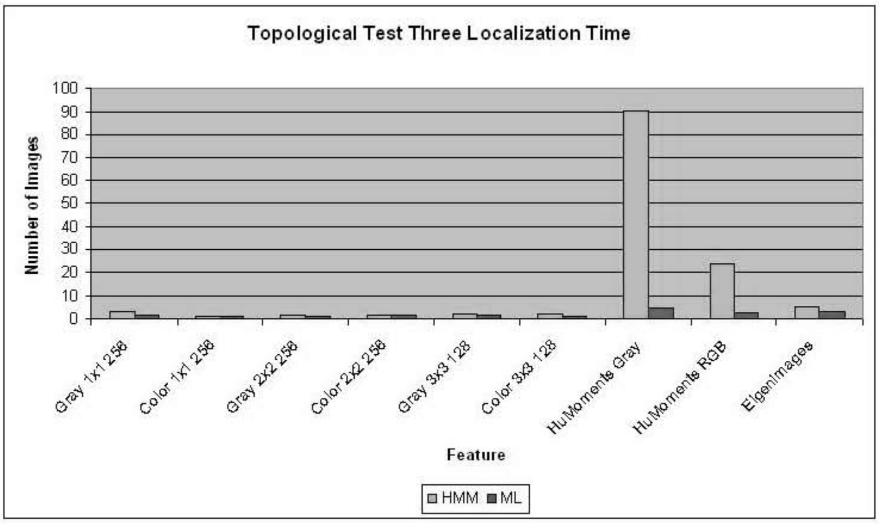

Fig. 15. Topological test three: localization time.

The number of images needed to be by the system gives an accurate assessment of how 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.

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 16 and 17 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 2. 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 was 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 3.

Both of these tests performed similar to those of the previous tests. Figures 18 to 21 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.

5. Conclusion

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 experiments. These experiments also show the ability of the system to recognize several locations that look very similar. The hallways, for example, Figures 5 and 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.

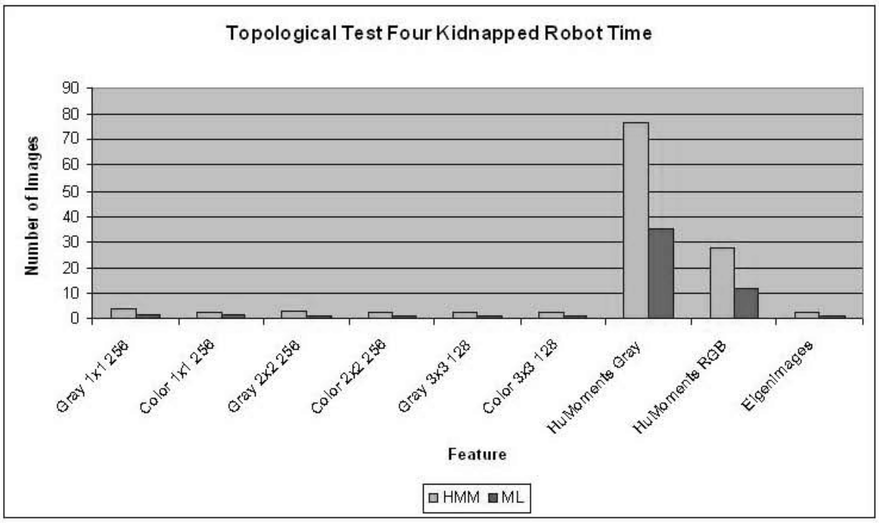

Fig. 16. Topological test four: kidnapped robot problem.

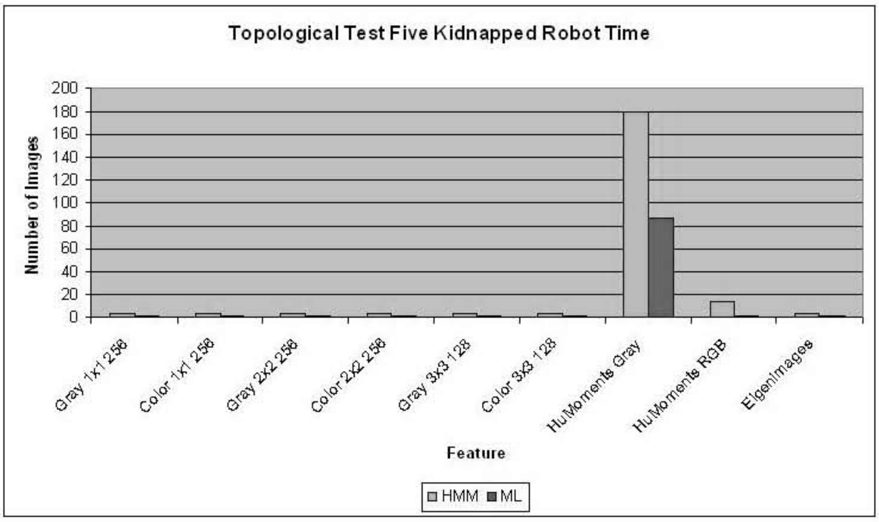

Fig. 17. Topological test five: kidnapped robot problem.

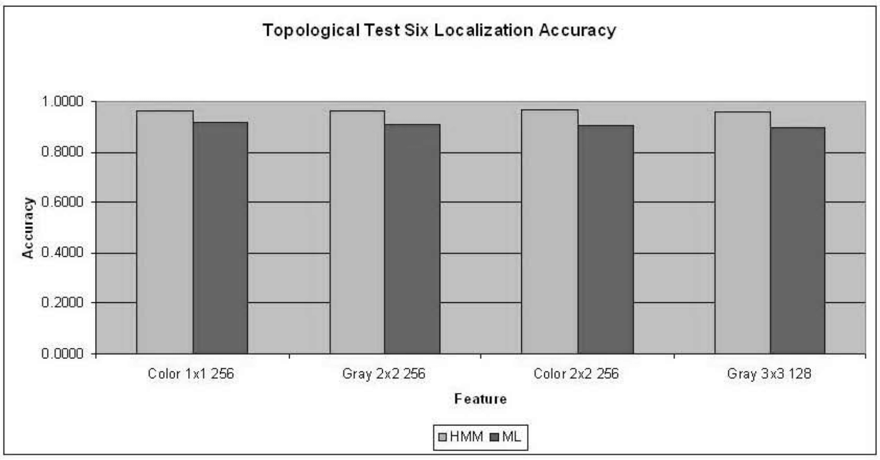

Fig. 18. Topological test six: position accuracy.

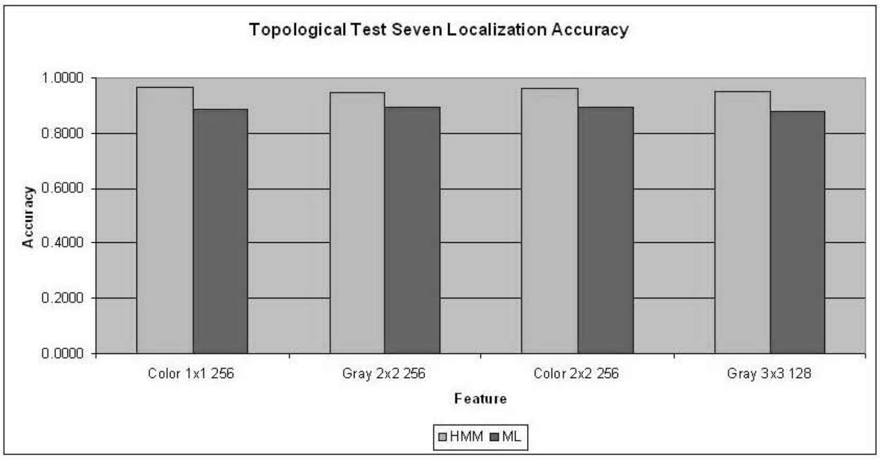

Fig. 19. Topological test seven: position accuracy.

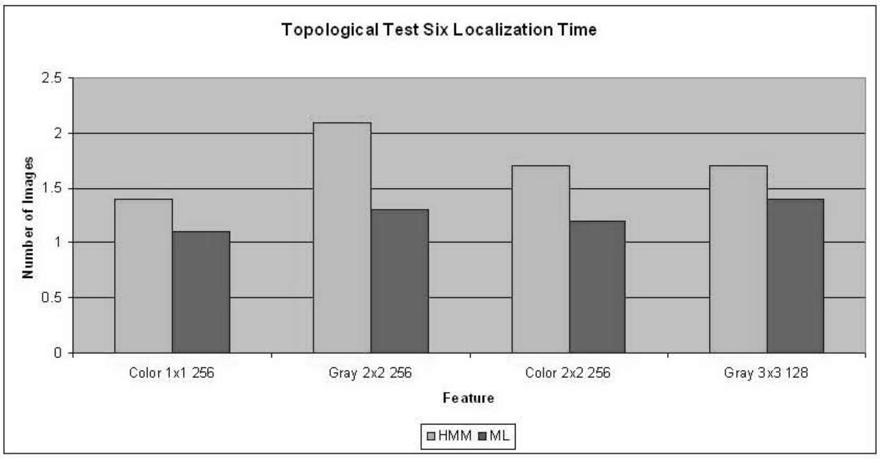

Fig. 20. Topological test six: localization time.

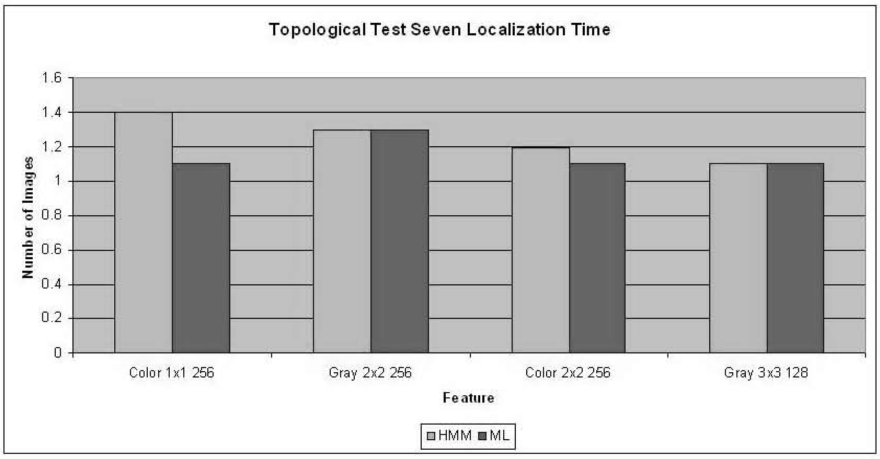

Fig. 21. Topological test seven: localization time.

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 system described determines only the location in a topological map. See [1] to how the system can be used to determine the location at a finer scale as well as the orientation.

5.1. 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.

5.2. 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.

This work was based on extracting a single feature from images. For environments such as in the polar regions, a single feature may not be sufficient. Work will also be done in determining if combining features results in better performance, especially in the polar regions where the images contain little texture.

Maps of larger environments, such as that of a college campus, will also be created to evaluate the system. The system could potentially be used in automobiles to guide drivers to parking lots near their buildings, or in self-driving automobiles.

5.3. PRISM/CReSIS Robotics

The localization system can also be used in a team based multiple robot scenario like the one described in [13]. The system could use the geometric localization to determine its location relative to the other robots. If a combination of sensors is 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 o or land.

For CReSIS robot [2], [4] to be completely autonomous. 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.

ACKNOWLEDGMENTS

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.

AUTHORS

Eric L. Akers - Department of Mathematics and Computer Science, Elizabeth City State University.

Arvin Agah* - Department of Electrical Engineering and computer Science, University of Kansas, Lawrence, KS 66045-7621 USA . E-mail: agah@ku.edu.

* Corresponding author

References

[1] Akers E.L., Agah A., „Large-scale localization using only a camera”. In: *Proceedings of the 2008 IEEE International Conference on Technologies for Practical Robotics Applications (TePRA)*, Woburn, Massachusetts, 2008, pp. 25-30.

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

[3] Akers E.L., Stansbury R.S., Agah A., „Long-term survival of polar mobile robots”. In: *Proceedings of the 4th International Conference on Computing, Communications and Control Technologies (CCCT)*, Orlando, Florida, vol. II, 2006, pp. 329-333.

[4] Akers E.L., Stansbury R.S., Agah A.,, Akins T.L., „Mobile robots for harsh environments: lessons learned from field experiments”. In: *Proceedings of the 11th International Symposium on Robotics and Applications (ISORA)*,

Budapest, Hungary. 2006, p. 16.

[5] Blaer P., Allen P. K., „Topological mobile robot localization using fast vision techniques". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington D.C., 2002, pp. 1031-1036.

[6] Bouman C.A., *„Cluster: An unsupervised algorithm for modeling Gaussian mixtures"*, 1997. Available from *http://www.ece.purdue.edu/~bouman*.

[7] Burgard W., Cremers A.B., Fox D., Hahnel D., Lakemeyer G., Schulz D., Steiner W., Thrun S., „The interactive museum tour-guide robot". In: *AAAI/IAAI*, 1998, pp. 11-18.

[8] Burgard W., Derr A., Fox D., Cremers A., „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, 1998, pp. 730-735.

[9] CReSIS. Center for Remote Sensing of Ice Sheets, 2007. URL: *http://www. cresis.ku.edu*.

[10] DARPA grand challenge, 2006. *http://www.darpa.mil/grandchallenge/index.asp*.

[11] Dempster A., Laird N., Rubin D., „Maximum likelihood from incomplete data via the em algorithm (with discussion)". *Journal of the Royal Statistical Society*, 1977, 39, pp. 1-38.

[12] Duda R.O., Hart P.E., Stork D.G. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2000.

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

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

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

[16] Jogan M., Leonardis A., Wildenauer H., Bischof H., „Mobile robot localization under varying illumination". In: *Proceedings of the International Conference on Pattern Recognition (ICPR)*, Quebec, Canada, 2002, pp. 1-4.

[17] Kosecka J., Li F., „Vision based topological markov localization". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004, vol. 2, pp. 1481-1486.

[18] Kr̆ose B., Vlassis N., Bunschoten R., Motomura Y., „A probabilistic model for appearance-based robot localization". *Image and Vision Computing*, 2001, vol. 19, no. 6, pp. 381-391.

[19] KU PRISM Team. Polar Radar for Ice Sheet Measurements, 2004. URL: http://www.ku-prism.org.

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

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

[22] Murase H., Nayar S.K., „Visual learning and recognition of 3-D objects from appearance". *International Journal of Computer Vision*, 1995, vol. 14, no. 1, p. 524.

[23] Nourbakhsh I., Powers R., Birchfield S., „Dervish an oce-navigating robot". *AI Magazine*, 1985, vol. 16, pp. 53-60.

[24] Porta J., Verbeek J., Krse B., „Active appearance-based robot localization using stereo vision". *Autonomous Robots*, 2005, vol. 18, no. 1, pp. 59-80.

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

[26] Russell S.J., Norvig P., *„Artificial Intelligence: A Modern Approach"*. Englewood Clis, New Jersey: Pearson Education, 2003.

[27] Sim R., Dudek, G., „Comparing image-based localization methods". In: *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003, pp. 1560-1562.

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

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

[30] Torralba A., Murphy K., Freeman W., Rubin M., „Context-based vision system for place and object recognition". In: *International Conference on Computer Vision Nice*, France, 2003.

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