Feature Generation For Robot Navigation and Scene Classification

Bijna Kizhakootil Balan

1.1 Introduction

The ability to navigate in its environment is important for any mobile robot. Apart from a purpose that relates to the navigation to specific places in its environment, focus should also be placed on avoiding dangerous situations such as collision with an obstacle and unsafe conditions. External sensors are essential for a mobile robot to respond to its environment and determine its position. Among the many requirements for external sensors, accuracy and a clear view are necessary for collision avoidance and faster motion. Ultrasonic sensors are one of the most popular sensors for indoor mobile robot which acquires environmental information with distances. Vision sensors are also getting used in mobile robots easily because of the recent improvements in cameras and processors. Omni directional images can provide segment information around the robot [3] .Our real world experiments were conducted on a physical robot (Adept Pioneer 3-DX) in in-door environment which comprises of 16 sonar sensors and an Omni-directional camera mounted on it. In order for the robot to make a better understanding of the environment, the perceived information should be effectively interpreted as relevant features. The aim of our study was to generate these relevant features from sonar readings for robot navigation and features from Omni-directional images for scene classification.

1.2 Experimental Design and Setup

The initial study was conducted on developing features from Sonar readings. The experiments were carried out on simulation environment developed by Dr. Frank Hoffmann. The entire experiment series on sonar readings were conducted on below mentioned behaviors

- Corridor Following
- Obstacle Avoidance
- Dead End

1.2.1 Graphical maps

Graphical maps were created for corresponding behaviors using Mapper3 software provided by the manufactures of Adept Pioneer 3dx robot. Various maps have been created to test corridor following, dead end and obstacle avoidance behavior. The following experiments were carried out on six maps - two maps for each behavior. Environments have been classified into seen and unseen. Seen Maps are used for taking the training data.Unseen maps are used for testing. The Unseen maps are comparatively more complex than seen maps. It is done intentionally to study the performance of the developed features under complex environment. The table 1.1 shows all the different maps used for the analysis.

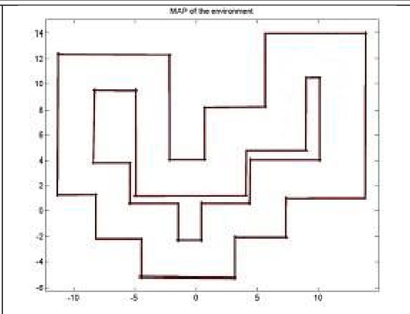
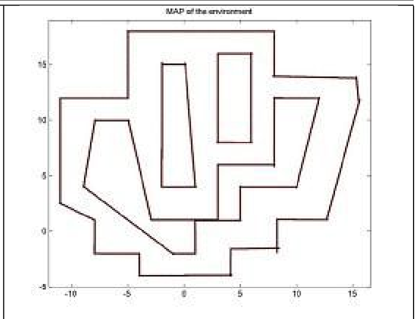
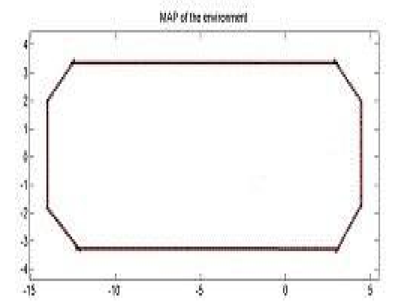
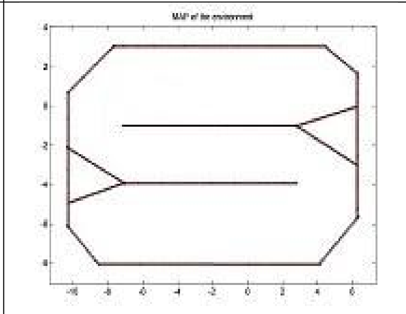
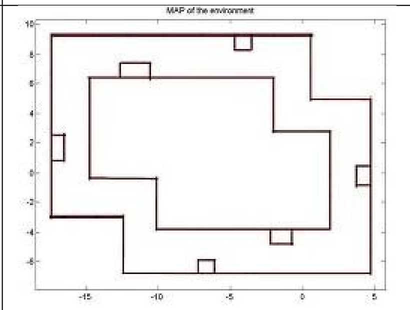
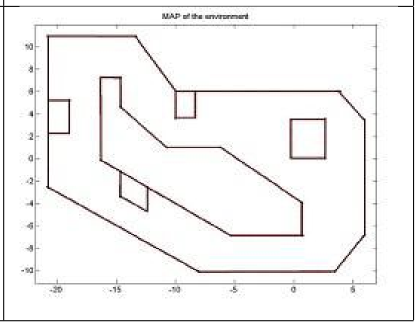
Behavior	Maps(Seen)	Maps(Unseen)
Corridor Following		
Dead End		
Obstacle Avoidance		

Table 1.1: Various maps used in analysis

1.2.2 Control Behavior

Control behavior used in our analysis and experiments has been developed by my teammate **??** which basically satisfies two conditions

1. Centering of the robot in the corridor
2. Obstacle avoidance

For satisfying the above mentioned conditions, the control behavior used the sonar readings from the front eight sensors of the robot. These readings have been divided into three zones, namely – Left zone, Right Zone and Critical Zone. The readings from left and right zones were used for centering and values from critical zone for obstacle avoidance.

1.2.3 Sonar Map

Since the simulation environment was not capable of providing us with segmented images, we created a binary segmented image with the sonar readings (Sonar map).

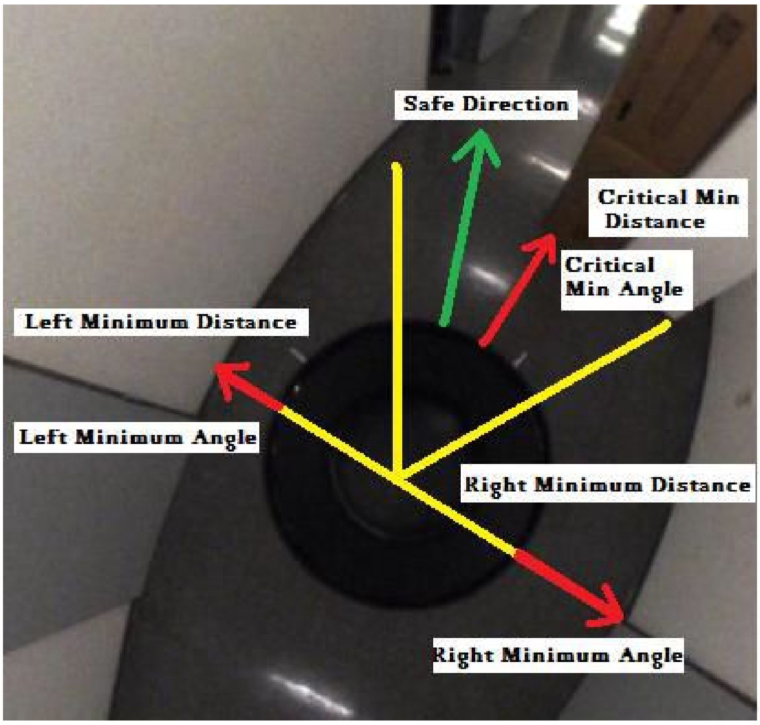


Figure 1.1: Figure showing zones and features

As the value given by each sonar depicts that there is no obstacle in the distance denoted, we considered an area formed from the values of 16 sensor readings, to be a free space around the robot. This was represented in the sonar map which gave us the free space segmentation region. The sonar map was created with the following assumptions.

1. Angle between two consecutive sonar sensors is a constant.
2. Since the area between two sonar readings is unknown, the presence of obstacle in this region was assumed to be in an ascending or descending manner (according to sonar values) between the preceding and succeeding sonar sensors.

Since our region of interest was restricted to the front area of the robot, the sonar map was generated with only front eight sonar sensors.

1.3 Features

Range information features were developed from sonar map. The features were developed by considering values from the front eight sonar sensors which was our region of interest. The region of interest was divided into three zones namely,

1. Left Zone (120° to 180°)
2. Right Zone (0° to 60°)
3. Critical Zone (60° to 120°)

In the figure 1.1 yellow line denotes area that separates each zone. Following features were generated from the sonar map.

1. **Left Minimum Distance ($D_{minLeft}$)**

   This gives the proximity of obstacle in the left zone. In the above figure 1.1, the red line in the left zone denotes the same. It is calculated as follows.

   $$D_{minLeft} = \frac{1}{min(\frac{d_i}{(1+cos\theta_i)})}$$

   where d_i is the distance to the obstacle at angle θ_i . The angle θ_i ranges from 120° to 180°.

2. **Left Minimum Angle (θ_{Left})**

   This gives the corresponding angle of proximity of obstacle in the left zone. In the above figure 1.1, θ_{Left} corresponds to the angle of the red line in the left zone.

3. **Right Minimum Distance ($D_{minRight}$)**

   This gives the proximity of obstacle in the right zone. In the above figure 1.1, the red line in the right zone denotes the same. It is calculated as follows.

   $$D_{minRight} = \frac{1}{min(\frac{d_i}{(1+cos\theta_i)})}$$

   where d_i is the distance to the obstacle at angle θ_i . The angle θ_i ranges from 0° to 60°.

4. **Right Minimum Angle (θ_{Right})**

   This gives the corresponding angle of proximity of obstacle in the right zone. In the above figure 1.1, θ_{Right} corresponds to the angle of the red line in the right zone.

5. **Critical Minimum Distance ($D_{minCritical}$)**

   This gives the proximity of obstacle in the critical zone. In the above figure 1.1, the red line in the critical zone denotes the same. It is calculated as follows.

   $$D_{minCritical} = \frac{1}{min(\frac{d_i}{(1+cos\theta_i)})}$$

   where d_i is the distance to the obstacle at angle θ_i . The angle θ_i ranges from 60° to 120°.

6. **Critical Minimum Angle ($\theta_{Critical}$)**

   This gives the corresponding angle of proximity of obstacle in the critical zone. In the above figure 1.1, $\theta_{Critical}$ corresponds to the angle of the red line in the critical zone.

7. **Safe Direction (θ_{Safe})**

   This gives the corresponding angle of maximum distance to the obstacle in the critical zone. In the above figure 1.1, θ_{Safe} is denoted by green line.

During the initial phase of the project, the zones were kept fixed, a detailed study was carried out on optimization of zones by my teammate **??**.

1.4 Feature Analysis

1.4.1 Initial Analysis with Feed Forward Network

An initial test was done along with the control behavior to analyze the range information features developed. The control parameters in the control behavior are linear velocity and angular velocity. Due to the non linearity in the relationship between the features and the control parameters, we used a simple feed forward neural network to check the ability of generated features to predict the control parameters. A neural network with 5 hidden neurons was used in the study, where the features developed were the input parameters and curvature was the output parameter.

$$Curvature = \frac{AngularVelocity}{LinearVelocity}$$

1.4.2 Optimizing the Feed Forward Neural Network

The motive behind this study was to figure out the best feed forward network suited for each behavior using newly generated range information features. The detailed study and results of Corridor Following and Obstacle Avoidance is included in the report of my teammate **??**. The results from Dead End behavior is included below. The analysis was done for feed forward network with hidden neurons ranging from 5 to 10. The neural network was trained with the developed range features as input and the curvature as the output. Two maps were considered in which one map was used to take the training data (seen map) and other map was used to evaluate the performance of the network in unknown conditions (unseen map). Ten different starting points were taken on seen map for collecting training data and the trained neural network was simulated with 20 different starting points (10 on seen map and 10 on unseen map). From the simulation data the following metrics were calculated.

1. **Average Normalized Mean Squared Error (Av. NMSE)**

   Normalized mean squared error between the predicted (neural network) and the actual target (output from control behavior) is calculated. The value is averaged over the total number of test runs (20 runs).

2. **Collision Percentage**

   Collision percentage gives number of times the robot collided out of total number of test runs (20 runs).

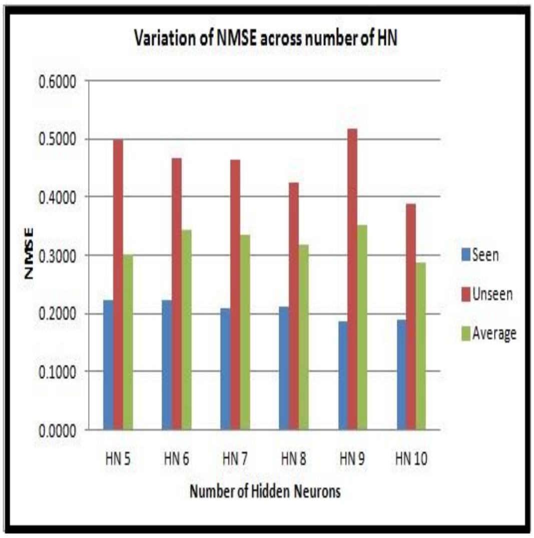
3. **Average Turn Rate (deg/s)**

   This gives the average turn rate predicted by the network over the total number of test runs (20 runs).

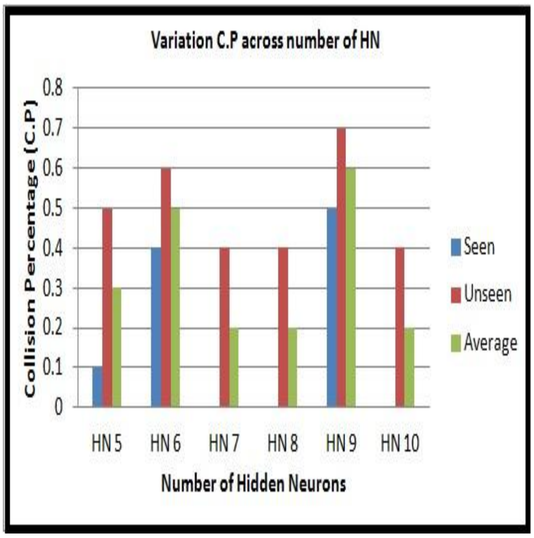
Behavior Analyzed – Dead End

Results for dead end behavior have been tabulated in table 1.2 and the corresponding bar diagrams.

Minimum number of hidden neurons is always better for a neural network to avoid over fitting. Among the different runs of experiments carried out on similar study, a slight difference was observed among corresponding values. A better network can be chosen by making a combined analysis of collision percentage and average NMSE



(a) Normalized Mean Squared Error



(b) Collision Percentage

Figure 1.2: Comparison of performance metric for various hidden neurons,Tab- 1.2

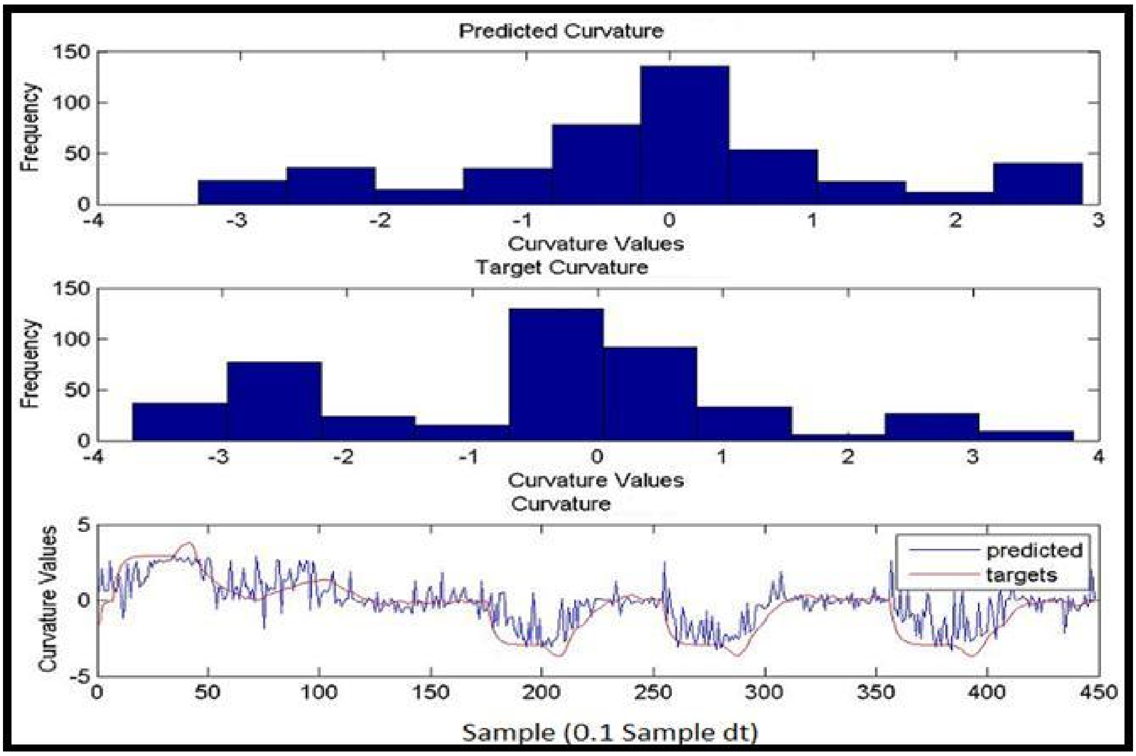


Figure 1.3: Figure showing the distribution and plot of curvatures, Table - 1.2

		HN 5	HN 6	HN 7	HN 8	HN 9	HN 10
Av. NMSE	Seen	0.2232	0.2214	0.2075	0.2112	0.1867	0.1891
	Unseen	0.4966	0.4655	0.4629	0.4233	0.5179	0.3875
	Average	0.3000	0.3435	0.3352	0.3172	0.3523	0.2883
Collision Per.	Seen	0.1	0.4	0	0	0.5	0
	Unseen	0.5	0.6	0.4	0.4	0.7	0.4
	Average	0.3	0.5	0.2	0.2	0.6	0.2
Av. Turn Rate	Seen	6.3253	8.5125	7.7543	6.7884	8.4322	6.6893
	Unseen	18.4532	13.8732	16.2389	11.9874	11.4642	14.7643
	Average	12.3893	11.1928	11.9966	9.3879	9.9482	10.7268

Table 1.2: Comparison of performance metric for various hidden neurons,Fig - 1.2

values. From the above results for this particular run of experiment, it can be observed that, a network with hidden neurons eight has better results when compared to others. For calculating the test error for each set of hidden neurons, runs with 10 different starting points in an unseen map is taken and an average of which is calculated as the Unseen NMSE value. For comparison, a figure 1.3 showing the distribution and plot of predicted and target curvatures of a single run (NMSE value - 0.4312) is included. A linear filter is applied in our control behavior which is not applied in the neural network. This causes some fluctuations in the predicted curvature values.

1.4.3 Optimizing the Feature Combination

The motive behind this study was to figure out the best feature combination suited for each behavior using newly generated range information features. The detailed study and results of Corridor Following and Obstacle Avoidance is included in the report of my team mate **??**.The results for Dead End behavior is included below.

The experimental set up and procedures were similar to that of the analysis explained in section 1.4.2. The analysis was done for feed forward network by keeping the hidden neurons constant (8 hidden neurons) across the run of the experiment. The neural network was trained with the different combinations of range features as input and the curvature as the output. The combinations of features were done logically. The following combinations were analyzed in the experiment.

1. $D_{minLeft}$,$D_{minRight}$,$D_{minCritical}$
2. θ_{Left}, θ_{Right},$\theta_{Critical}$
3. $D_{minLeft}$,$D_{minRight}$,$D_{minCritical}$,θ_{Safe}
4. θ_{Left}, θ_{Right},$\theta_{Critical}$,θ_{Safe}
5. $D_{minLeft}$,$D_{minRight}$,$D_{minCritical}$, θ_{Left}, θ_{Right},$\theta_{Critical}$
6. $D_{minLeft}$,$D_{minRight}$,$D_{minCritical}$, θ_{Left}, θ_{Right},$\theta_{Critical}$, θ_{Safe}

From the simulation data the following metrics were calculated

1. **Average Normalized Mean Squared Error (Av. NMSE)**

2. **Collision Percentage**

3. **Average Turn Rate (deg/s)**

Behavior Analyzed – Dead End

The corresponding analysis has been carried out with two types of training data. First one was similar to that of explained in section 1.4.2. (Ten different starting points on seen map for training data and the test run with 20 different starting points). The second consisted of sparse training data (Five different starting points on seen map for training data and the trained neural network was simulated with 20 different starting points). This was done to analyze the impact of less training data on different feature combination. Results for two types of training data have been tabulated separately.

Results with first set of training data

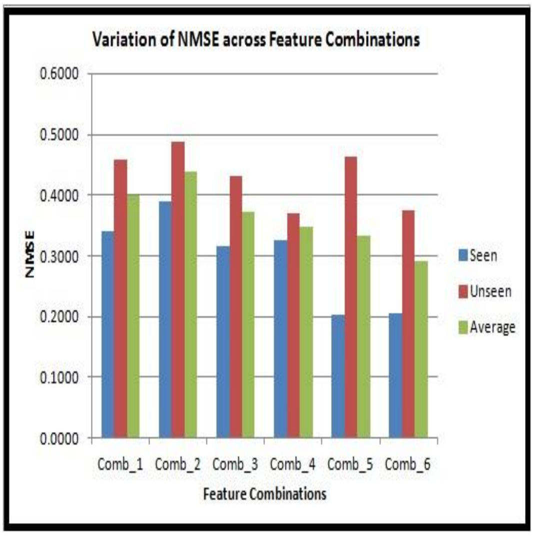
		Comb1	Comb2	Comb3	Comb4	Comb5	Comb6
Av. NMSE	Seen	0.3398	0.3902	0.3148	0.3265	0.2022	0.2064
	Unseen	0.4577	0.4871	0.4314	0.3695	0.4631	0.3757
	Average	0.3987	0.4386	0.3731	0.3480	0.3327	0.2911
Collision Per.	Seen	0.3	0.1	0.2	0.1	0.5	0
	Unseen	0.2	0.6	0.5	0.2	0.5	0.2
	Average	0.25	0.35	0.35	0.15	0.5	0.1
Av. Turn Rate	Seen	8.0284	6.6851	6.7915	7.9862	8.3141	8.0389
	Unseen	9.4280	6.6402	6.7912	8.7221	7.8023	9.8104
	Average	8.7282	6.6626	6.7914	8.3541	8.0582	8.9246

Table 1.3: Comparison of performance metric for various feature combinations,Fig-1.4

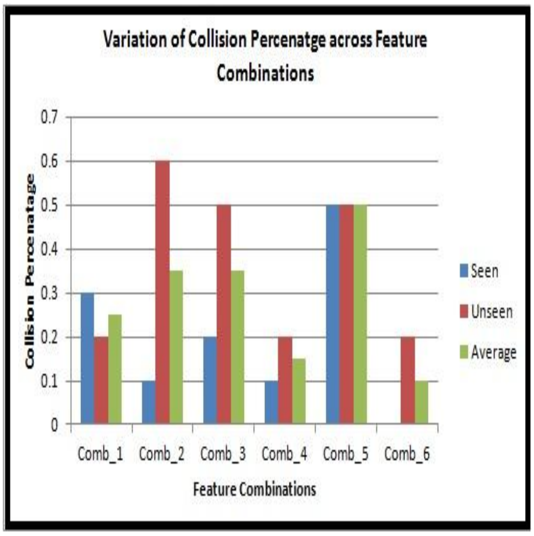
Among the different runs of experiments carried out on similar study, a slight difference was observed among corresponding values. A better combination of features was chosen by making a combined analysis of collision percentage and average NMSE values. From the results shown in table 1.3 and corresponding bar diagrams, for this particular run of experiment, it can be observed that, a combination of θ_{Left}, θ_{Right},$\theta_{Critical}$,θ_{Safe} has better results when compared to others. For calculating the test error for each feature combination, runs with 10 different starting points in an unseen map is taken and an average of which is calculated as the Unseen NMSE value. For comparison, a figure 1.5 showing the distribution and plot of predicted and target curvatures of a single run (NMSE value - 0.3800) is included. A linear filter is applied in our control behavior which is not applied in the neural network. This causes some fluctuations in the predicted curvature values.

Results with sparse training data

Among the different runs of experiments carried out on similar study, a slight difference was observed among corresponding values. A better combination of features can be chosen by making a combined analysis of collision percentage and average NMSE



(a) Normalized Mean Squared Error



(b) Collision Percentage

Figure 1.4: Comparison of performance metric for various feature combinations,Tab-1.3

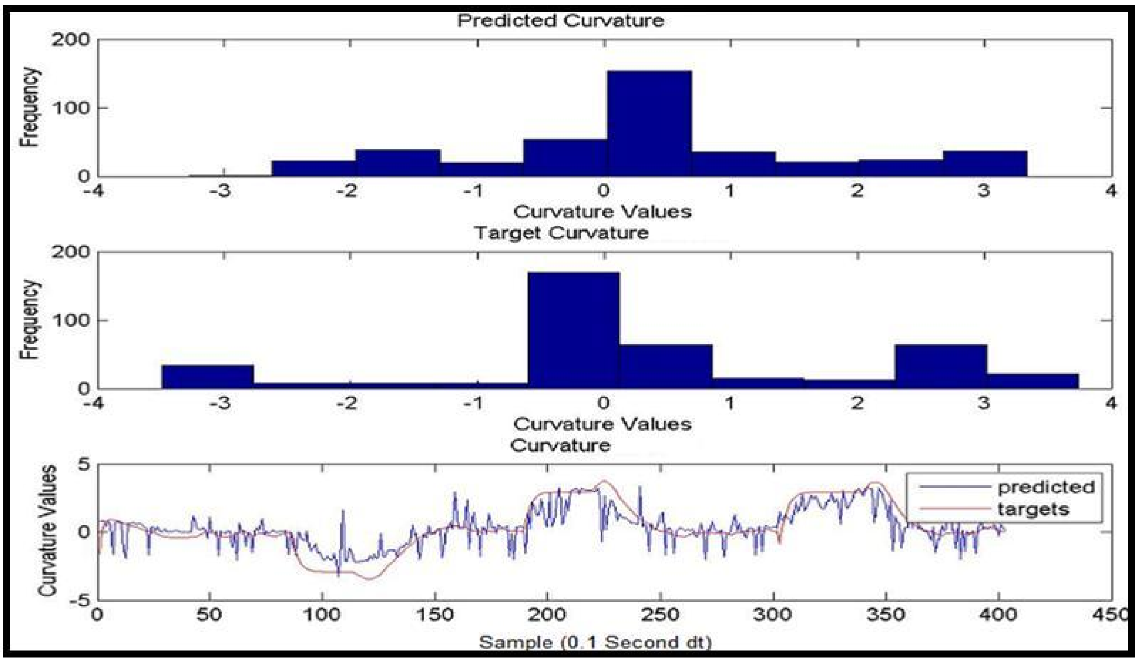
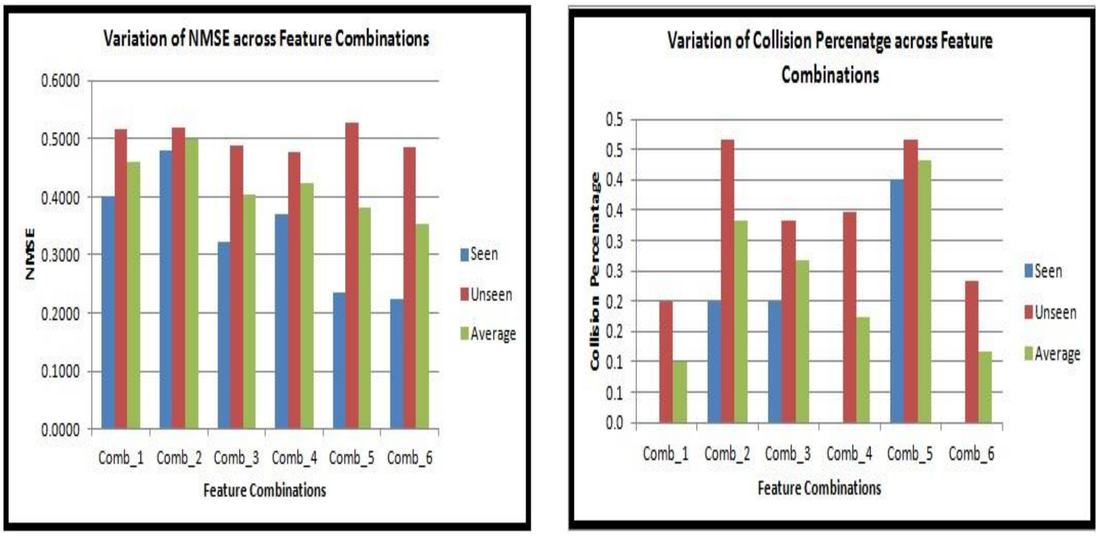


Figure 1.5: Figure showing the distribution and plot of curvatures, Table - 1.3



(a) Normalized Mean Squared Error (b) Collision Percentage

Figure 1.6: Comparison of performance metric for sparse data, Table - 1.4

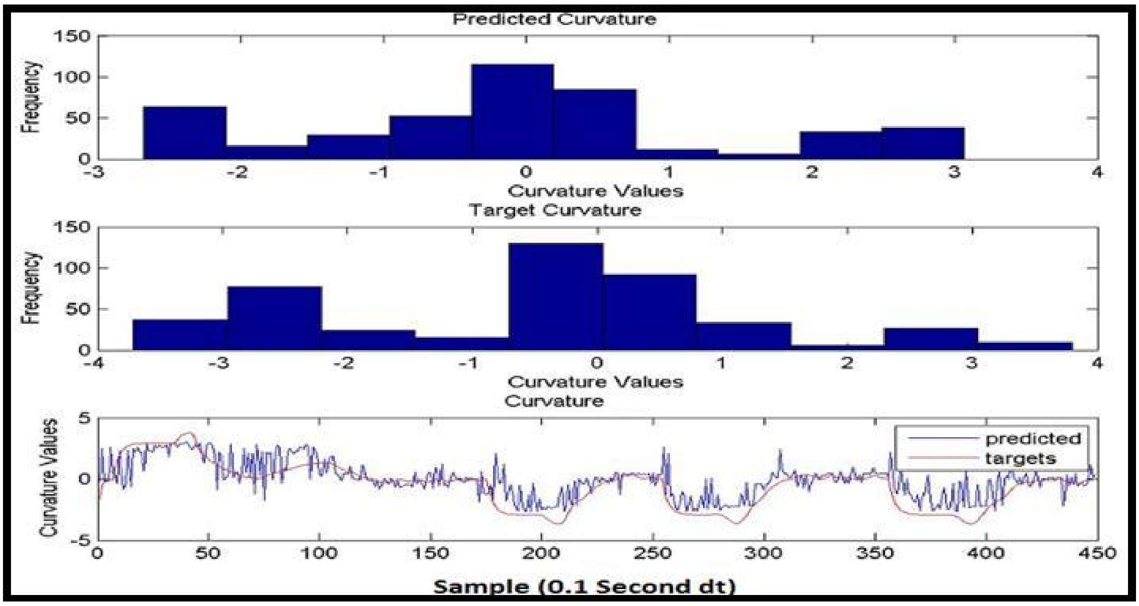


Figure 1.7: Figure showing the distribution and plot of curvatures, Table - 1.4

		Comb1	Comb2	Comb3	Comb4	Comb5	Comb6
Av. NMSE	Seen	0.4017	0.4804	0.3221	0.3690	0.2350	0.2238
	Unseen	0.5174	0.5205	0.4872	0.4763	0.5276	0.4855
	Average	0.4595	0.5004	0.4047	0.4227	0.3813	0.3546
Collision Per.	Seen	0	0.2	0.2	0	0.4	0
	Unseen	0.2	0.5	0.3	0.3	0.5	0.2
	Average	0.1	0.3	0.3	0.2	0.4	0.1
Av. Turn Rate	Seen	10.5200	7.3840	11.7701	4.0840	3.9945	7.0998
	Unseen	16.8617	11.4602	15.6538	7.1307	6.6280	12.2543
	Average	13.6909	9.4221	13.7119	5.6074	5.3112	9.6770

Table 1.4: Comparison of performance metric for sparse data, Figure - 1.6

values. From the results shown in table 1.4 and the corresponding bar diagrams, for this particular run of experiment, it can be observed that, the combination of θ_{Left}, θ_{Right},$\theta_{Critical}$,θ_{Safe} has better results when compared to others. For calculating the test error for each feature combination, runs with 15 different starting points in an unseen map is taken and an average of which is calculated as the Unseen NMSE value. For comparison, a figure 1.7 showing the distribution and plot of predicted and target curvatures of a single run (NMSE value - 0.4592) is included. A linear filter is applied in our control behavior which is not applied in the neural network. This causes some fluctuations in the predicted curvature values.

1.5 Raw Image features for Scene classification

Unknown indoor environment is one of the main challenges in the navigation operation of the mobile robot. Visual sensors arise as an emerging tendency for such purpose due to the richness of the information they provide about the surrounding environment. Based on a single omnidirectional camera mounted on the real robot, we can exploit the maximized horizontal field of view, there by generating a robust representation of the environment and providing us with more information. The data used in the experiments consists of a large sets of omnidirectional images captured in two different scenarios – Rooms and Corridors. We carried out a set of analysis in images of real indoor environments to test the prediction accuracy. Experimental results demonstrate the prediction percentage based on test images from real-time.

1.5.1 Visual Features

The visual features considered during the study extracts the relevant information directly from the Omni directional images captured. The features considered are the following

1. **Entropy**

   Entropy of the segmented image tells if the floor to the front is clear or clustered with obstacle. It is given by the equation,

$$Entropy = -\sum(f log_2 f)$$

where f is the histogram count of the pixel values.

2. **Edge pixel Ratio**

   The edges are found from the Omni directional image using canny edge detector and the ratio of edge pixels available in the whole image is found.

$$EdgepixelRatio = \frac{Numberofedgepixels}{Totalnumberofpixels}$$

3. **Axis Ratio**

   An ellipse is fit into the edge image, from which major principal axis (the longest axis of an ellipse) and minor principal axis (the shortest axis of an ellipse) is calculated. Then the ratio these axes are taken.

$$AxisRatio = \frac{Majorprincipalaxis}{Minorprincipalaxis}$$

1.5.2 Classification using Gaussian Mixture Model

Classification, as a part of unsupervised learning problem, has always been of attention for its various applications. Also, many methods are brought forward to tackle this problem. Here GMM is used for unsupervised learning because it can dig out various data patterns and cluster together those data that share similar behaviors [1] .The parameters of the Gaussian mixture model were initialized using K-Means clustering based on the maximum Likelihood (ML) estimation using Expectation Maximization (EM). In order to minimize the number of components in the Gaussian mixture model, thereby limiting the problem of over fitting data, Bayesian information Criteria (BIC) was used. Thus an optimized model with better result and less complexity is obtained. The Probability Density Function (PDF) of a multivariate Gaussian represented by means and covariance matrix was calculated which was used as a deciding factor among the test data.

1.5.3 Training Data set

Images of different corridors and rooms across various departments were collected with the help of our Demonstration team. Also additional image databases of corresponding environments were taken from internet for training as well as testing [2] .1500 images each for room and corridor were used for training. Comparison of feature values for the training data has been done, which is shown in figure 1.8.

With the shown training data set, optimized Gaussian mixture model was developed. The model representations with various combinations of features are shown in figure 1.9.

1.5.4 Test Data Analysis

A total of 1000 images were used in the testing phase in which the distribution is as follows

1. Corridor – 600 Images

2. Room – 400 Images

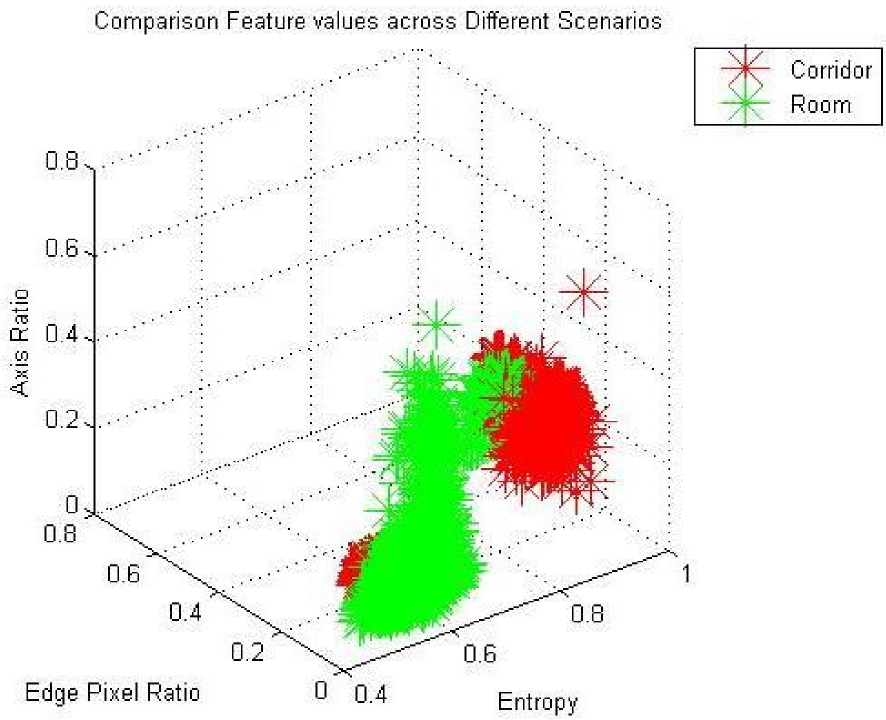


Figure 1.8: Figure showing the training data set

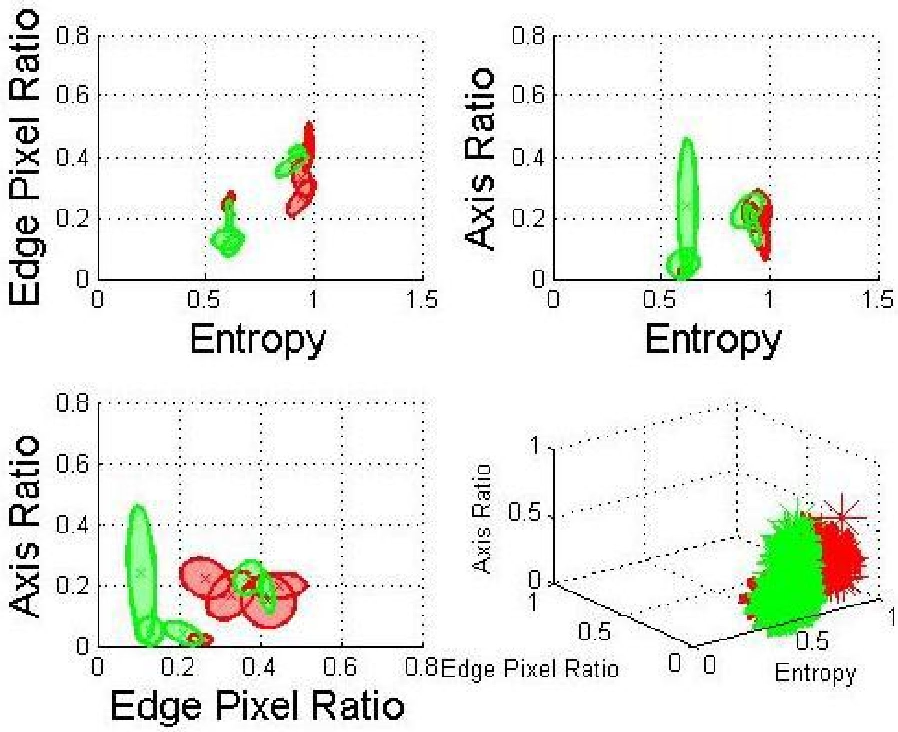


Figure 1.9: GMM representation in 2D along with training data set

The analysis was carried out in three phases and the corresponding results were tabulated using a confusion matrix.

Individual Analysis of Test Images

Each of the available images were analyzed individually and the prediction of the scene was analyzed. The results are shown in table 1.5.

		Prediction Class		
		Corridor	Room	Unidentified
Actual Class	Corridor	578	20	2
	Room	150	242	8

Table 1.5: Confusion matrix of individual image prediction

From the table 1.5, the prediction percentage of each scene as well as the overall prediction percentage was calculated as shown in table 1.6.

Prediction Percentage	
Corridor	0.96
Room	0.61
Overall	0.82

Table 1.6: Prediction percentage

Analysis on Image Sequences

This study was conducted by considering different sequence of consecutive images. The total probability for the entire sequence of images was found and then the scene was predicted using GMM. This was done to check whether there is a chance to increase the prediction percentage when compared to that of individual image prediction. The result is shown in table 1.7.

	Number of Images						
		5	10	25	50	75	100
Consecutive Combinations	10	0.84	0.83	0.83	0.82	0.82	0.80
	30	0.823	0.803	0.793	0.793	0.783	
	50	0.814	0.790	0.792	0.794	0.798	

Table 1.7: Prediction percentage on image sequence

The results in the table 1.7 depicts the information of prediction percentage for each set of image sequences chosen for particular number of times from the available test image database. The results shows that choosing 5 consecutive images in a sequence gives a marginal rise in the prediction percentage when compared to other sets as well as individual image prediction.

Prediction based on Voting Scheme

Apart from the type of study done previously, a different approach was carried out on the image sequence. The prediction of each image in the sequence was done separately and then a voting scheme was applied. The scene which has the higher number of votes was predicted as the scene for the entire image sequence. This will help to discard the outliers if present. The corresponding results are shown in table 1.8.

		Prediction Class		
		Corridor	Room	Unidentified
Actual Class	Corridor	597	3	0
	Room	115	285	0

Table 1.8: Confusion matrix of voting scheme prediction

Since the study in section 1.5.4 shows that an image sequence with 5 images gives better results and hence this was considered for the study in this section. The above table shows the prediction percentage of each scene (predicted on image sequence with 5 images through voting scheme). The prediction percentage of each scene as well as the overall prediction percentage was calculated as shown in table 1.9.

Prediction Percentage	
Corridor	0.995
Room	0.713
Overall	0.882

Table 1.9: Prediction percentage for Voting scheme

Since there was a considerable improvement in the prediction percentage, this method was chosen.

1.6 Conclusion

A detailed study on generating relevant features from sonar readings for robot navigation and features from Omni-directional images for scene classification has been done. As a result, a set of features has been developed. The generated features were analyzed through different experiments and tests. The better results obtained from these investigations show that the corresponding features generated can be used for robot navigation and scene classification with much accuracy. Further research and improvements can be done on this to obtain more accurate and reliable operations.

References

[1] M. Cao, "Practice on classification using gaussian mixture model," 2010.

[2] (). Cosy localization database, [Online]. Available: `http://www.cas.kth.se/COLD/downloads.php`.

[3] H. Everett, *Sensors for Mobile Robots: Theory and Application.* A. K Peters, 1995.