Advanced Materials Research Vols. 403-408 (2012) pp 3559-3569
Online available since 2011/Nov/29 at www.scientific.net

doi:10.4028/www.scientific.net/AMR.403-408.3559

Collision Avoidance in Unstructured Environments for Autonomous Robots: A Behavioural Modelling Approach

Chika O. Yinka-Banjo[1], Isaac O. Osunmakinde[2], Antoine Bagula[1]
[1]Department of Computer Science, University of Cape Town, Rondebosch, Cape Town South Africa
[2]Mobile Intelligent Autonomous Systems, Modelling and Digital Sciences Department Council for Scientific and Industrial Research (CSIR), South Africa
Emails: chikagog@gmail.com, iosunmakinde@csir.co.za, bagula@cs.uct.ac.za

Keywords: Collision Avoidance, Unstructured Environment, Autonomous Robots, Behavioural Model, Modelling and Simulation

Abstract. Collision avoidance is one of the important safety key operations that needs attention in the navigation system of an autonomous robot. In this paper, a Behavioural Bayesian Network approach is proposed as a collision avoidance strategy for autonomous robots in an unstructured environment with static obstacles. In our approach, an unstructured environment was simulated and the information of the obstacles generated was used to build the Behavioural Bayesian Network Model (BBNM). This model captures uncertainties from the unstructured environment in terms of probabilities, and allows reasoning with the probabilities. This reasoning ability enables autonomous robots to navigate in any unstructured environment with a higher degree of belief that there will be no collision with obstacles. Experimental evaluations of the BBNM show that when the robot navigates in the same unstructured environment where knowledge of the obstacles is captured, there is certainty in the degree of belief that the robot can navigate freely without any collision. When the same model was tested for navigation in a new unstructured environment with uncertainties, the results showed a higher assurance or degrees of belief that the robot will not collide with obstacles. The results of our modelling approach show that Bayesian Networks (BNs) have good potential for guiding the behaviour of robots when avoiding obstacles in any unstructured environment.

Introduction

Robotics is the engineering, science and technology, design, manufacture, application and structural disposition of robots [16]. Making progress towards autonomous robots is a major practical interest in a wide variety of application areas including manufacturing, construction, mining, medical surgery and assistance for the disabled and the aged. The basic characteristic of an autonomous robot is its capability to operate/navigate independently in an unknown, known or partially-known environment [2]. To achieve this level of robustness, some methods need to be developed to provide collision-free navigation for robots in an unstructured environment. An unstructured environment in this work is a type of environment that has no specific pattern and where obstacles are static.

In our autonomous robot Collision Avoidance Model (CAM), safety measures need to be put in place in order to make autonomous robots avoid colliding with obstacles while navigating to achieve their goal. Fig. 1 shows an ongoing key challenge that has a red-coloured object with four wheels as the robot [15], and the chairs as the obstacles. This is usually a repetitive process of moving to a new position, sensing the environment, calculating the distances and taking action to the next level based on the information gathered from the environment. Most of the difficulties faced in these processes originated from the nature of the real world, an unstructured environment and environmental uncertainties[3]. For instance, any prior knowledge about the environment is, in general, incomplete, uncertain and approximate [7]. For example, perceived information is usually unreliable, stable features in the environment may change with time and agents can modify the environment. Fig. 1, a robotic vehicle set to avoid collisions with the chairs in an unstructured indoor environment.

Many studies to date have focused on improving the navigation system of autonomous robots. Hongjun et al. [6], proposed a novel method for sensor planning using mobile robot localization based on Bayesian Network inference. In their work, they proved that an autonomous robot cannot always determine its unique situation by local sensing information only. The reason is that, the sensor is prone to errors and a slight change of robotic behaviour deteriorates the sensing result. Jasmin et al. [7] describes how soft computing methodologies such as fuzzy logic, genetic algorithm and the Dempster-Shafer theory of evidence can be implemented in a mobile robot navigation system by using a reasoning and search system. In addition, Lazkano et al. implemented a doorcrossing behaviour in a mobile robot within an environment with smooth walls and doors using only sonar readings [4].

Using alternative modelling, this paper focuses on using the Bayesian Networks for investigating the collision avoidance task. This model captures uncertainties from the unstructured environment in terms of probabilities and performs reasoning with the probabilities. The reasoning in this case is the ability of the robot to learn the unstructured environment through the learning capabilities of the Bayesian Network [4]. The reasoning algorithm centres around Bayes' rule for calculating the posterior probability that a robot takes an action given data (obstacles distances). That is, we want to deal with expressions of the form:

$$\Pr(\text{Robot's behaviour?} \mid \text{Obstacles distances}). \quad (1)$$

The major contributions in this paper are as follows:

1)The application of Bayesian Network for building behaviour for mobile robots as a collision avoidance model in unstructured environments where obstacles are static.

2)Accounting for inevitable uncertainties embedded in unstructured environments as a way for making timely and accurate avoidances of obstacles through three experiments.

The remainder of this paper is arranged as follows. In Section 2, we present the theoretical background of the CAM as a class of Bayesian Network (BN) model. Section 3 presents the experimental setup of the proposed approach. The results of the three experimental applications and evaluations of the model are given in Section 4. Finally, conclusions and further work are given in Section 5.

Figure 1. A robotic vehicle set to avoid collisions with the chairs in an unstructured indoor environment.

Theoretical Background

Bayesian Networks. Probabilistic graphical models represented by directed acyclic graphs that have nodes as variables and arcs that show the conditional (in)dependencies among the variables [4] are Bayesian Networks. BN has two main components: the graphical structure and the conditional probabilities associated to each node of the network. These components can be established by the human expert who takes advantage of his knowledge about the relations among the variables. It can also be built automatically by implementing any automatic learning algorithm and finally it can be a combination of mixing the expert's knowledge and the learning mechanism [4]. In this experimental task, the BBNM is built using GeNle [19]. GeNle is a software that has the capability of building graphical networks with some automatic learning algorithms. For example, the Naive Bayes' classifier technique. The various ways in which a Bayesian Network can learn are: (i) known structures with complete data, (ii) known structures with incomplete data, (iii) unknown structures with complete data and (iv) unknown structures with incomplete data [14]. Figs. 2a and 2b illustrate how BN model is learnt from complete data as is the case for this experimental work. The pre-processing step known as discretization is performed by partitioning the possible values of continuous attributes into small number of inter-values, where each interval is mapped to a discrete symbol [8]. Discretization is applied whenever continuous data needs to be transformed into discrete data for effective feature construction and ease of modelling.

Bayesian Network Inference. The fundamental idea of solving a probabilistic network, BN, is to exploit the structure of the knowledge base (Database) to reason efficiently (inference) about the events and decisions of the problem domain, taking the inherent uncertainty into account [1]. The BN model in this work is achieved using the Naive Bayes. The Naive Bayes' model is most commonly used for classification because of its low model complexity and high computational power [1]. Considering Fig. 3, the node to be inferred is the class or query node, A. The evidence nodes are independent of each other and are dependent on the class node, A.

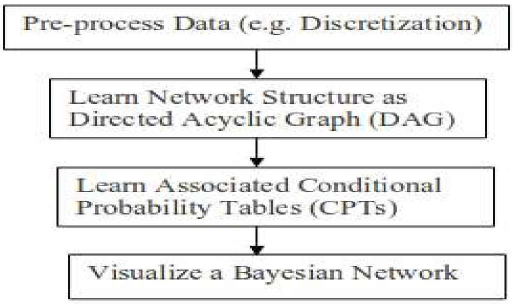

Figure 2 (a). Initial Stages of BN Learning

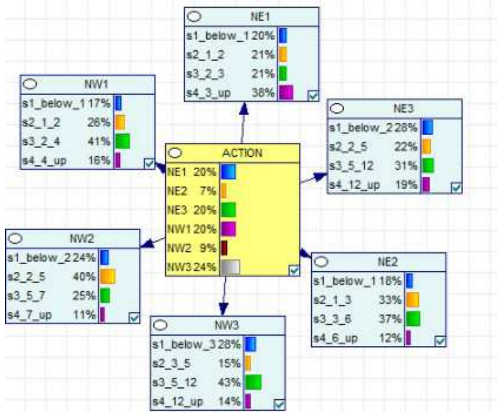

Figure 2 (b). Final Stage of BN Learning

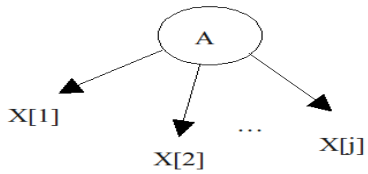

Figure 3. Representation of Bayes Net for Inference.

The only parent in the network is A. To calculate the BN inference using the network in Fig. 3 [20], we apply the Bayes' theorem as follows:

$$P\,(A|X[i],\ldots,X[j]) = \frac{P\,(X[i],\ldots,X[j]|A)P(A)}{P\,(X[i],\ldots,X[j])}, \quad (2)$$

where P (A|X[i],…, X[j]) is the posterior or degree of belief , A is obtained after obtaining the behaviour of the robot. It is called the original degree of belief when the likelihood and the prior are combined. The term P (X[i],…,X[j]|A) is the likelihood function of X[i] given A. It is taken as the probability of what we know, given what we don't know. P(A) is the prior and is called prior because it is the probability of A before making any observation or any inference [13] and P(X[i],…,X[j]) is the probability of data. Availability of data is an advantage to estimate the prior and conditional probability distribution P(A) and $P(X_1|A),\ldots,P(X_j|A)$ from data.

The Proposed Behavioural Model

In order to design a Behavioural Model, it is necessary to have a playerstage. Playerstage is a prototype of an unstructured environment that describes the robot's position and other parameters like obstacles.

The System Model for the CAM. The system model comprises three essential components which are: simulation of an unstructured environment, Behavioural Bayesian Network model (BBNM) and predictions by testing the model for obstacle avoidance. The first component of the system model discovered the system knowledge of obstacles distances which are used in the second component of the system model to build the Behavioural Bayesian Network Model. The last component reasons with the model by predicting the behaviour of a robot given obstacles distances in an unstructured environment.

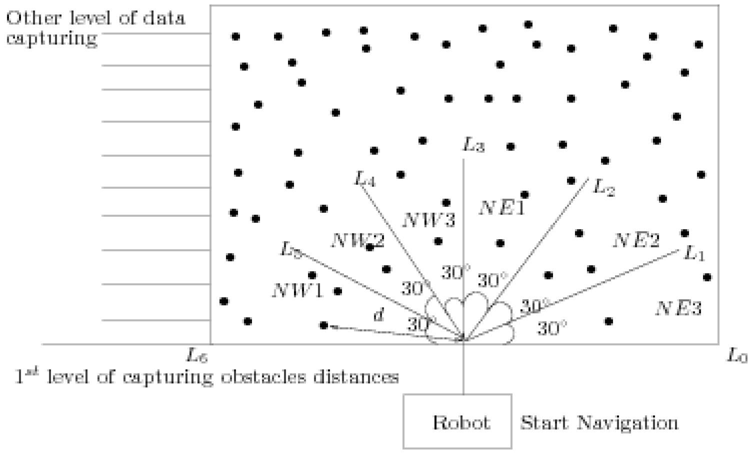

Figure 4 (a). Playerstage Prototype for Robot's Navigation

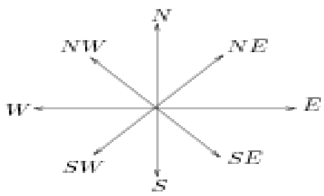

Figure 4 (b). Cardinal Directions for guiding the movements

Data Acquisition from Simulation. At first, data was collected from the environment for the construction of the network model. This was achieved by automating a code that generates points to represent obstacles as depicted in Fig. 4a. The unstructured environment the robots navigate is a representation of a real life indoor environment simulated as a playerstage. The implementation is a simple forward-looking movement as shown in Fig. 4a.

Furthermore, basic assumptions are made in generating data for the model(cardinal points, obstacles, etc.). Further details of the assumptions are detailed in the components of the playerstage below.

Components of the Playerstage

1) Cardinal Directions: To determine the geographical orientation of an autonomous robot, at each position, the notion of the cardinal direction is used. There are four major cardinal directions, or cardinal points. North (N), south (S), east (E) and west (W) and four main intermediate directions, north-east (NE), north-west (NW), south-west (SW) and south-east (SE) as depicted in Fig. 4b.

2) Robot: This is the initial position of the robot as depicted in Fig. 4a in the unstructured environment before navigation. The robot perceives the environment using its Light Detection and Ranging (lidar) sensor. The lidar sensor captures the closest obstacles' distances at the six geographical directions of the path. Out of all the closest obstacles distances captured, the obstacle with the farthest distance is picked. The robot navigates towards the farthest obstacle observation and reasons for the next direction. The new position of the robot forms the basis of its new direction. The process is repeated, it perceives the environment, checks for the closest obstacles distances in the next level of perception, picks the highest obstacle distance among the perceived obstacles distances and navigates towards it. It then reasons again for the next direction. The robot follows the above steps until it gets to its desired destination.

3) Obstacles: The obstacles are represented with random variables or random points, which correspond to the position of chairs, tables, etc. in the real life environment.

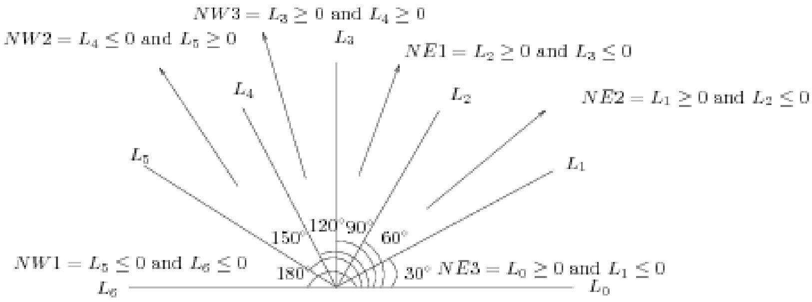

Figure 5. Bottom up obstacles direction capturing using some line conditions

4) Distance: To calculate the distances between the robot position and the obstacles, the Euclidean distance [21] or metric distance is adopted. Euclidean distance is the distance between two points. In the playerstage, d denotes the distance between the robot and the nearest obstacle. The obstacles' distances are calculated as shown in (3).

$$d=\sqrt{(x1-x2)^2 + (y1-y2)^2} \quad (3)$$

where

- x1 = robot's position at x-cordinate
- y1 = robot's position at y-cordinate
- x2 = obstacle's position at x-cordinate
- y2 = obstacle's position at y-cordinate

5) Lines: To partition the path into different geographical directions, we use the equation of a line. In our playerstage, we have six different lines captioned L0, ..., L6. The lines divide the path into six equal directions captioned NW1,...,NE3. as depicted in Fig. 4a. Fig. 5 shows how obstacles directions are captured. To capture the nearest obstacle distance to the robot in NE1 direction, the parameters in the NE1 partition are considered. These directions form the columns in the generated database. To capture the position and partition of the obstacles, equations of lines are also used. The lines are calculated as follows:

$$L_i = y - m_i x, \quad (4)$$

where $i = 1, \ldots, 6$ and m_i is the slope of line i.

6) Angles: The angles are calculated by uniformly dividing the path (180°) into a number of n-equal partitions (lines). In the playerstage, each partition has angle 30° as a result of dividing the path into 6-equal partitions.

$$\text{Angle} = \frac{180^\circ}{6} = 30^\circ \quad (5)$$

7) Level: Each navigation position is described as a level of obstacle perception at every instance in time. This level represents the rows in the generated database. Suppose from the robot's position, the obstacle distance at a particular position is represented by dij , where i ranges from 1,...,6 and j ranges from 1,...,n. In a general notion, the generated data will be represented as follows: LOP represents the levels of obstacles' perceptions and the column headings (NW1, . . . ,NE3) represent the geographical orientation of obstacles' distances and their positions. Therefore, each cell in the table represents the obstacles' distance measured. For example, in Fig. 6, d11 and d23 represent measured obstacles' distances as perceived by the robot in the geographical orientation NW1 and NW3, respectively. The Action column represents the highest distance position for each level of perception. That is:

	NW1	NW2	NW3	NE1	NE2	NE3	Action
LOP1	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}	d_{16}	NW3
LOP2	d_{21}	d_{22}	d_{23}	d_{24}	d_{25}	d_{26}	NW1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
LOPn	d_{n1}	d_{n2}	d_{n3}	d_{n4}	d_{n5}	d_{n6}	NE2

Figure 6. Data Generation Table

TABLE I. PSEUDO-CODE FOR DATA GENERATION AT PATH 180°

input:	*Obstacles in unstructured environment, U.*
output:	*Database of obstacles' distances, D(U).*
step 1:	*Initial position of robot at origin (0,0) on path (line) 180° .*
step 2:	*Set robots lidar sensor perceptions at level i(i =1,..., n) to m equal partitions of path P° each.*
step 3:	*Measure and record the nearest obstacles' distances di 's in each partition.*
step 4:	*Get the highest obstacle distance max (di 's).*
step 5:	*Action: Set neighbourhood threshold control parameter di' as the new robots's position.*
step 6:	*Repeat steps 1 to 5 until the robot navigates to its desired destination.*

$$\text{Action} = \max(\text{LOP}_i) \quad i = 1, \ldots, \text{n}. \tag{6}$$

For example,

For row LOP1:

$$d_{13} > (d_{11}, d_{12}, d_{14}, d_{15}, d_{16}) \Rightarrow d_{13} = \max(d_{ij})$$

In LOP1 and LOP2, we assumed the highest obstacle distance is NW3 and NW1, respectively. The pseudo-code in Table 1 summarises the steps involved in generating data for the CAM

The Behavioural Bayesian Network Model (BBNM). Once the network structure is obtained and the probability tables are calculated, the network model is ready for prediction [4]. The structure shows that there is only one parent or class node called Action and six variable/evidence nodes called NW1,…,NE3. The model also shows that the class node is conditionally dependent on the evidence nodes. This means the evidence can be propagated to get the posterior distribution. This is achieved by updating the posterior of the class variable after setting the evidence of each node. Propagation can be performed using exact methods or approximate methods. Exact methods calculate the exact posterior probabilities of the variables and this is usually the case where the network is simple. Some examples of exact methods are: variation elimination, clique tree propagation etc. In the case of complex network structures, approximate methods are used. Examples of approximate methods are: clustering, sampling etc. These methods use Bayes' rule for computation.

Scoring and Validation. We considered the K-fold cross-validation technique in this paper. With K-fold cross-validation, a single subsample of the known data is set aside as validation data for testing the model, and the remaining K-1 subsamples are used as training data [18]. We repeat the cross-validation process K times where each K subsamples are used exactly once as the validation data.

Experimental Evaluations and Results

One of the objectives of this paper is to bring the Behavioural model to practice with an emphasis on robotic applications and collision avoidance strategy. This consequently alleviates the robot's behaviour as it reasons over environmental uncertainties. To justify the universality of the CAM and to assure that our Behavioural modelling design is reproducible, different unstructured environments are used to test our model and implementation. Fig. 7 shows the proposed Behavioural model structure learned from Fig. 6 using the GeNle software [19]. Table 2 shows the steps involved in building this BBNM from data *D*. Note that the model obtained is a typical type of Naive Bayes as explained in Section 2.

TABLE II. PSEUDO-CODE FOR BEHAVIOURAL BAYESIAN NETWORK MODEL

input:	*Database of obstacle's distances, D(U).*
output:	*Behavioural Bayesian Network Model (BBNM).*
step 1:	*Discretize data D(U); Dis(U).*
step 2:	*Learn network structure from Dis(U) as Directed Acyclic Graph (DAG).*
step 3:	*Learn associated conditional probability tables (CPTs), say K_i, from Dis(U).*
step 4:	*Visualize Bayesian Network (BN) structure.*

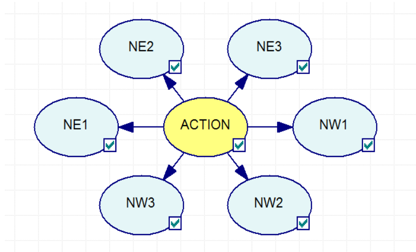

Figure 7. Behavioural Bayesian Network Model

Three experiments were conducted on unstructured environments with static obstacles including:
1) Performance accuracy of the collision avoidance model by plying modelled environment (ME).
2) Performance accuracy of the collision avoidance model by plying new directions.
3) Performance accuracy of the collision avoidance model by plying new unstructured environment (UE).

The performance accuracies of the CAM on each of these environments are also computed using the cross-validation technique [17] summarized in each of the experiments. With five level of obstacle perceptions (LOP) selected on each validation experiment, Figs. 8, 10 and 11 show the results of the expected robot behaviour (ERB) compared with the predicted robot behaviour (PRB). The validation results realised after the comparison are visualized on Tables III, IV and V. The performance accuracy of the model is computed as T from Figs. 8, 10 and 11 as expressed in (7). For dataset1 in Fig. 8 for example, T = 5/5*100% = 100%. The average performance accuracy is computed by finding the average of the number of validation experiments. In Table III, the average performance accuracy for experiment I is (100%+100%+100%)/(3) = 100%.

$$T = \frac{\text{Number of correct predictions}}{\text{Total number of cases}} \times 100\% \qquad (7)$$

Experiment I: Performance Accuracy of the CAM by Plying ME. Obstacles' distances are perceived by the robot's lidar sensor and measured. Fig. 8 shows the data used for experiment I. The objective of this experiment is to test the model's accuracy in the modelled environment. That is, some of the samples used for building the model are used to test the model's performance. Table III shows the results of the average performance obtained from the evaluation of the model.

To better measure the accuracy of the obtained model, 3-fold cross-validation is applied to the data and the average accuracy measure from the testing of the 3-folds is reported in Table III.

TABLE III. 3-FOLD CROSS-VALIDATION TABLE FOR EXPERIMENT I

3-Fold Cross-validation	Subsample	Precision
Validation Data 1	Dataset1	100%
Validation Data 2	Dataset2	100%
Validation Data 3	Dataset3	100%
Average Cross-Validation: 300/3= 100%		

	Percepts of Obstacle Distances							
E_i	**NW1**	**NW2**	**NW3**	**NE1**	**NE2**	**NE3**	**ERB**	**PRB**
E_1	5	8.71	1.83	6.14	0.34	0.44	NW2	NW2(91.0%)
E_2	2	4.92	4.09	5.8	1.91	3.65	NE1	NE1(90.6%)
E_3	10.51	7.02	5.07	3.98	0.06	0.05	NW1	NW1(94.0%)
E_4	5.1	5.91	5.95	4.01	2.16	0.18	NW3	NW3(72.0%)
E_5	1.65	5.54	5.82	3.21	7.27	3.02	NE2	NE2(78.3%)
Test Dataset 1: Accuracy = 5/5 = 100%								
E_i	**NW1**	**NW2**	**NW3**	**NE1**	**NE2**	**NE3**	**ERB**	**PRB**
E_1	6.54	9.8	2.48	6.1	0.65	0.42	NW2	NW2(92.7%)
E_2	2.72	8.13	5.7	5.35	4.17	1.91	NW2	NW2(67.5%)
E_3	0.22	0.24	1.38	2.21	3.53	4.79	NE3	NE3(88.0%)
E_4	3.07	5.67	5.39	5.72	0.44	0.38	NE1	NE1(43.3%)
E_5	0.19	0.57	4.67	6.29	9.95	2.76	NE2	NE2(81.8%)
Test Dataset 2: Accuracy = 5/5 = 100%								
E_i	**NW1**	**NW2**	**NW3**	**NE1**	**NE2**	**NE3**	**ERB**	**PRB**
E_1	9.41	10.68	2.81	5.91	0.31	0.62	NW2	NW2(83.2%)
E_2	3.75	5.82	6.09	1.05	1.25	0.77	NW3	NW3(78.4%)
E_3	4.49	4.88	5.82	1.57	5.41	0.72	NW3	NW3(93.3%)
E_4	2.84	4.61	5.67	6.35	2.58	3.5	NE1	NE1(69.1%)
E_5	1.02	2.23	3.4	4.49	8.02	3.75	NE2	NE2(74.9%)
Test Dataset 3: Accuracy = 5/5 = 100%								
Average Accuracy = 100%								

Figure 8. Data from Modelled Unstructured Environment

The charts in Fig. 9 show the predicted behaviour of the robot when navigating in the modelled environment represented by dataset1 in Fig. 8. The charts show the proportion of the degree of belief for each evidence (NW1,…,NE3) of obstacles' distances in dataset1. The area of each chart is proportional to the quantity of the degree of belief. Observe that the largest portion of each chart in Fig. 9 represents the behaviour of the robot for each set of obstacles perceived. This tallies with the degree of beliefs in the PRB column of dataset1 in Fig. 8.

The highest posterior probability of the class node (Action) is used to select the behaviour of the robot. For example, NW2 is the highest obstacle distance at E1 of dataset1. At this point, the robot navigates away from the closest obstacles' distances and moves towards the farthest obstacle distance NW2. When approaching, it perceives obstacles' distances again at a new position and reasons for the next direction. Equation 9 expresses how the predictions of the robot behaviour is obtained for NW1 of dataset1 in Fig. 8 and (8) is the predictor variable. The collision avoidance is a continuous process as the robot perceives new obstacles' distances as evidence over time. Looking at the results in Fig. 8 and Table III, the model performance is 100% accurate. This shows that there is a 100% guarantee of collision-free navigation for autonomous robots in the modelled environment.

$$Pr(A=NW1|NW1=5, NW2=8.71, \ldots, NE3=0.44). \tag{8}$$

Using the Bayes' rule described in Section 2, equation 8 becomes

$$\frac{Pr(NW1=5, \ldots, NE3=0.44|A=NW1) \times Pr(A=NW1)}{Pr(NW1=5, \ldots, NE3=0.44)}. \tag{9}$$

More information on (9) is available in [9].

Experiment II: Performance Accuracy of the CAM by Plying New Directions. For experimental verification, the BBNM is deployed in a different environment where robots navigate in new directions e.g. diagonal movements. This tests for increased uncertainty on the robots' navigation because the test dataset here were not used for building the model. We performed validation experiments similar to those in experiment I with the data generated from the simulation of diagonal movements. The evaluation of the model is also performed using the 3-fold cross-validation technique. The results are tabulated in Fig. 10. Performance accuracy for experiment II using 3-fold cross validation is shown in Table IV. The overall performance accuracy result obtained during the robot's navigation in this direction looks promising as shown in Fig. 10 and Table IV.

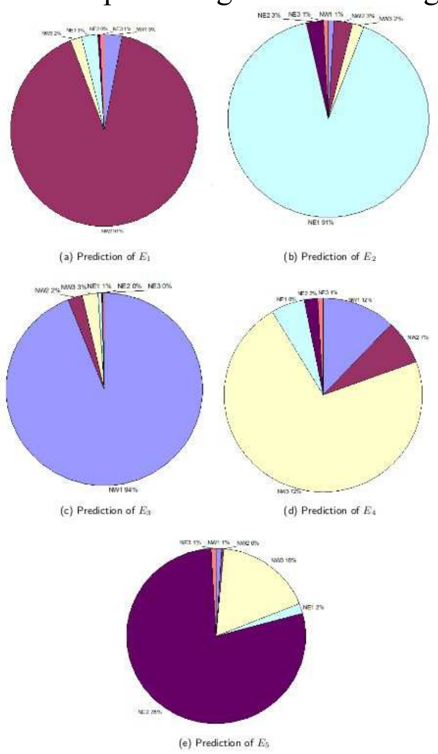

Figure 9. Pictorial Results of PRB of Dataset1 in Fig. 8

TABLE IV. 3-FOLD CROSS-VALIDATION TABLE FOR EXPERIMENT II

3-Fold Cross-validation	Subsample	Precision
Validation Data 1	Dataset1	60%
Validation Data 2	Dataset2	80%
Validation Data 3	Dataset3	80%
Average Cross-Validation: 220/3= 73.3%		

	Percepts of Obstacle Distances							
E_i	NW1	NW2	NW3	NE1	NE2	NE3	ERB	PRB
E_1	4.85	6.51	5.85	4.06	0.56	2.51	NW2	**NW3(43%)**
E_2	4.48	3.63	5.49	0.61	1.07	3.09	NW3	NW3(46.6%)
E_3	0.65	0.15	0.76	0.98	0.74	5.55	NE3	NE3(38.2%)
E_4	4.11	4.76	5.74	3.02	4.35	3.77	NW3	**NE2(39.8%)**
E_5	3.58	4.99	3.53	3.19	9.59	2.49	NE2	NE2(52.8%)
Test Dataset 1: Accuracy = 3/5 = **60%**								
E_i	NW1	NW2	NW3	NE1	NE2	NE3	ERB	PRB
E_1	2.75	2.53	6.41	7.46	7.74	3.21	NE2	NE2(37.5%)
E_2	2.47	3.6	1.07	3.48	7.88	4.23	NE2	**NE1(30.2%)**
E_3	2.12	2.71	0.49	4.75	5.21	5.81	NE3	NE3(47.8%)
E_4	3.85	1.82	6.42	3.07	2.88	2.88	NW3	NW3(44.9%)
E_5	0.72	0.6	0.77	0.41	3.38	3.87	NE3	NE3(33.2%)
Test Dataset 2: Accuracy = 4/5 = **80%**								
E_i	NW1	NW2	NW3	NE1	NE2	NE3	ERB	PRB
E_1	7.53	1.66	0.87	0.5	0.69	0.67	NW1	NW1(40.8%)
E_2	6.31	6.12	7	5.74	4.62	2.07	NW3	**NW2(24.9%)**
E_3	0.54	0.41	0.63	0.31	1.46	7.32	NE3	NE3(46.7%)
E_4	0.56	0.79	1.86	1.59	2.54	4.75	NE3	NE3(65.5%)
E_5	3.01	4.35	3.66	5.45	1.52	1.92	NE1	NE1(59.2%)
Test Dataset 3: Accuracy = 4/5 = **80%**								
Average Accuracy = **73.3%**								

Figure 10. Data from New Direction

	Percepts of Obstacle Distances							
E_i	NW1	NW2	NW3	NE1	NE2	NE3	ERB	PRB
E_1	4.4	3.73	1.91	4.89	2.42	2.74	NE1	NE1(42.2%)
E_2	6.99	4.26	1.22	4.35	5.13	2.29	NW1	NW1(26.0%)
E_3	1.62	3.93	4.01	4.96	4.48	5.9	NE3	**NW3(22.5%)**
E_4	5.78	5.35	4.82	2.22	0.15	0.08	NW1	NW1(31.6%)
E_5	4.61	6.4	3.07	2.57	0.84	0.9	NW2	NW2(64.4%)
Test Dataset 1: Accuracy = 4/5 = **80%**								
E_i	NW1	NW2	NW3	NE1	NE2	NE3	ERB	PRB
E_1	0.07	0.54	1.85	3.87	2.87	9.55	NE3	NE3(72.3%)
E_2	1.46	1.02	3.05	2.49	3.79	3.12	NE2	**NE3(30.3%)**
E_3	0.06	0.02	3.08	1.81	5.4	9.59	NE3	**NE3(75.7%)**
E_4	3.43	6.35	4.1	2.09	1.1	3.98	NW2	NW2(73.3%)
E_5	5.11	3.76	2.12	2.03	0.08	0.1	NW1	NW1(36.5%)
Test Dataset 2: Accuracy = 4/5 = **80%**								
E_i	NW1	NW2	NW3	NE1	NE2	NE3	ERB	PRB
E_1	7.27	3.51	5.26	4.97	0.6	0.07	NW1	NW1(65.8%)
E_2	3.12	4.58	4.75	3.24	3.57	2.71	NW3	**NE1(60.5%)**
E_3	0.92	1.75	3.3	3.52	8	5.44	NE2	NE2(72.4%)
E_4	0.84	0.89	1.45	4.46	2.7	8.37	NE3	NE3(62.6 %)
E_5	2.98	7.26	2.47	1.4	1.77	0.91	NW2	NW2(39.4%)
Test Dataset 3: Accuracy = 3/5 = **60%**								
Average Accuracy = **73.3%**								

Figure 11. Data from New Unstructured Environment

TABLE V. 3-FOLD CROSS-VALIDATION TABLE FOR EXPERIMENT III

3-Fold Cross-validation	Subsample	Precision
Validation Data 1	Dataset1	80%
Validation Data 2	Dataset2	80%
Validation Data 3	Dataset3	60%
Average Cross-Validation: 220/3= 73.3%		

Experiment III: Performance Accuracy of the CAM by Plying New Unstructured Environment. In this section, we carry out the last experiment to further evaluate the performance of the model. This provides the experimental results for an obstacle avoidance model when an autonomous robot navigates in a new environment while perceiving new obstacles' distances. Fig. 11 and Table V show the average performance obtained for three different validations conducted in the new environment. In each validation, obstacles distances are randomly sampled and selected from the new test dataset. Each set contains five evidence (E1,…,E5) rows of perceptions and six obstacles' distances (NW1,…,NE3) as columns. The ERB column which denotes the expected robot behaviour offers the prior knowledge of how the robot should behave from the simulated experiment. The PRB column which denotes the predicted robot behaviour also shows the result of the model. After each experiment, the result of the ERB and that of PRB are compared and evaluated. The essence of this is to test how accurate our model will perform when the robot finds itself in a new environment where patterns/knowledge of the obstacles are unknown. This is achieved and the results in Fig. 11 and Table V proved the adaptability of the model in such an unknown environment.

Conclusion and Future Work

In this paper, we have presented the Collision Avoidance Model (BM) to improve the safe navigation of an autonomous robot in an unstructured environment where obstacles are static. The results of the CAM are promising and are able to predict the behaviour of the robot in an unstructured environment. The results of experiment I shows that there is certainty in the degree of belief that the robot will not collide with any obstacle in that environment. The 73.3% accuracy achieved from experiment II shows that the model is able to adapt to an unstructured environment with increased uncertainties. The BBNM has also been tested in a new unknown environment in experiment III where uncertainties are more and the results show the potentials of the model as promising to cope with unstructured environments. However, the purpose of the proposed Behavioural model is to investigate its capability to handle uncertainties for robot to navigate freely in any unstructured environment. The experimental results obtained reveal this achievement. Having achieved a level of certainty in the degree of belief for the proposed model, we are working to make our idea robust and flexible by carrying out investigations on (i) dynamic obstacles; (ii) moving from a specified start position to goal position; (iii) comparism with other models e.g. Hopfield Neural Network, etc; and (iv) test our idea on the field.

References

[1] B. K. Uffe, and L.M. Anders, Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis, Springer, 2008.

[2] Y.C. Kim, S.B. Cho, and S.R. Oh, “Map-building of a real mobile robot navigation with GA.,” International Journal of Fuzzy System., vol. 4. 2002, pp.696–703.

[3] G. Orioli, G. Ulivi and M. Vendittelli, “Real-time map building and navigation for autonomous robots in robots in unknown environments,” IEEE Transactions on System, Man and Cybernetics part B., vol.3. 2002, pp. 316-333.

[4] E. Lazkano, B. Sierra, A. Astigarraga, and J.M. Martinez-Otzeta, “On the use of bayesian networks to develop behaviours for mobile robots,” Robotics and Autonomous Systems, vol. 55. 2007, pp. 253-265.

[5] J. Gasos, and A. Rosetti, “Uncertainty representation for mobile robots: Perception, modelling and navigation in unknown environments.,” Fuzzy Sets and Systems., vol. 107, 1999, pp 1-24.

[6] Z. Hongjun, and S. Shigeyuki, “Mobile robot localization using active sensing based on Bayesian network inference,” Robotics and Autonomous Systems, vol. 55. 2007, pp. 292-305.

[7] V. Jasmine, L. Bakir, and P. Branislava,” A 3-level autonomous mobile robot navigation system designed by using reasoning and search approaches,” Robotics and Autonomous Systems, vol.54, 2006, pp. 989-1004.

[8] J. Alipio, T. Luis, B. Pavel, C. Rui, and G. Joao, “Knowledge Discovery in Databases,” 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal, PKDD 2005 .

[9] S. Russell, and P. Norvig, Artificial Intelligence Artificial Intelliegnce Artificial Intelliegnce: A Modern Approach 2nd Edition. Prentice Hall Series Inc. New Jersey, 2003.

[10] Stroud, K.A., Engineering Mathematics: Sixth Edition, Palgrove Macmillan, 2007.

[11] L.A. Zadeh, and J. Kacprzyk, “Fuzzy Logic for the Management of Uncertainty,” John Wiley and Sons, New York, 1992.

[12] H. Asoh, and Y. Motomura, “Combining probabilistic map and dialog for robust life-long office navigation,” International Robots and System, 1996, pp. 880 – 885.

[13] D. Fox, W. Burgard, S. Thrun, and A.B. Cremers,” A hybrid collision avoidance method for mobile robots,” Proc. IEEE Inter. Conference on Robotics and Automation, 1998.

[14] I.O. Osunmakinde, Telecommunications Networks Fraud Detection using Bayesian Networks. AIMS 2005.

[15] http://www.l3s.de/morob/evaluation1.html:Evaluation of different Robots platforms in educational applications.

[16] http:/www.en.wikipedia.org/wiki/Robotics : Robotics.

[17] I.O. Osunmakinde, and T. Ndhlovu, “Ground plane detection for autonomous robots in complex environments inclined with flexed far-field terrains,” Proc. Of 14th IASTED Inter. Conference on robotics and applications (RA 2009), Cambridge, Massachusetts, USA. ACTA Press, pp. 664-095, 2009.

[18] http://www.en.wikipedia.org/wiki/Cross-validation : Cross-validation.

[19] http://genie.sis.pitt.edu : Genle and Smile.

[20] www.cs.huji.ac.il/~nirf/Nips01-Tutorial: Learning Bayesian. Network from data.

[21] http://www.en.wikipedia.org/wiki/Euclideandistance:Euclidean Distance

MEMS, NANO and Smart Systems
10.4028/www.scientific.net/AMR.403-408

Collision Avoidance in Unstructured Environments for Autonomous Robots: A Behavioural Modelling Approach
10.4028/www.scientific.net/AMR.403-408.3559