Motion Prediction of Moving Objects in a Robot Navigational Environment Using Fuzzy-based Decision Tree Approach

Received 10th January 2010; accepted 5th July 2010

Vijay S. Rajpurohit, M.M. Manohara Pai

Abstract:
In a dynamic robot navigation system the robot has to avoid both static and dynamic objects on its way to destination. Predicting the next instance position of a moving object in a navigational environment is a critical issue as it involves uncertainty. This paper proposes a fuzzy rule-based motion prediction algorithm for predicting the next instance position of moving human motion patterns. Fuzzy rule base has been optimized by directional space approach and decision tree approach. The prediction algorithm is tested for real-life bench- marked human motion data sets and compared with existing motion prediction techniques. Results of the study indicate that the performance of the predictor is comparable to the existing prediction methods.

Keywords: *short term motion prediction, fuzzy rule base, rule base optimization, fuzzy predictor algorithm, directional space approach, decision tree approach.*

1. Introduction

For an autonomous mobile robot, performing a navigation-based task in an unknown environment to detect and avoid encountered obstacles is an important issue. It is also a key function for the robot body safety, as well as for the task continuity. Generally, the architecture for the vision-based robotic systems with the ability of obstacle detection and avoidance are relatively complicated. This may be attributed to the extraction of information from a stream of the site images consisting of the static and dynamic obstacles. In a dynamic robot navigation system, the robot has to acquire the information on moving objects and predict their future positions in order to make path planning efficient. Short term object motion prediction in a dynamic robot navigation environment refers to the prediction of next instance position of a moving object based on the previous history of its motion. The living beings and vehicles characterize the dynamic environment and exhibit motion in various directions with different velocities.

Real-life data often suffer from inaccurate readings due to environmental constraints, sensors, size of the objects and possible change in motion pattern of the moving objects. This needs the system to be robust to handle these uncertainties and predict next instance object position as accurate as possible within a short duration. As a result, object motion prediction still continues to be an active field of research. Research literature has addressed solutions to the short term object motion predictions with different methods such as: curve fitting or regression methods [7], [18], neural network based approaches [1], [2], [4], Hidden Markov stochastic models [19], Bayesian Occupancy Filters [5], Extended Kalman Filter [9], [12], Stochastic prediction model [17], regression methods [18], [7] proposed in the literature, sample the positions of moving object at definite time intervals, and fit the information to the regression equation. With the current sampling positions, the regression model predicts the position of the object for the next sampling duration. The main drawback of this method is the estimation of model coefficients in real-life environment, which makes the system complex. Amalia Foka *et al.* [1],[2] have proposed a Polynomial Neural Network (PNN) architecture for object motion prediction. The PNN uses a second order polynomial equation as a transfer function at each node. Training is done using evolutionary method. The algorithm needs huge amount of data sets for training and the performance of the algorithm is poor in case of unseen datasets. Relative Error Back Propagation neural network [4] for object motion prediction considers rectilinear motions of moving objects. The algorithm needs huge dataset for training and quality of results depend on the training data set used. Statistical methods for estimating obstacle locations using statistical features have been proposed such as Hidden Markov Model [19] to predict object motion. The method is computationally intensive. The method proposed by R. Madhavan *et al.* [12] uses Extended Kalman Filter. Each prediction step is dependent on the previous sequence of observations made and the quality of prediction reduces with increase in time and space horizon. C. Laugier and S. Petti [5] have proposed Baysean programming framework to predict the future position of moving object. The navigational environment is represented as a four dimensional occupancy grid. The method is not suitable for large scale environment because of intrinsic complexity and numerical computations. R. Irajit *et al.* [15] in their work have proposed a methodology based on Artificial Potential Fields (APF) method which provides simple and effective motion planners for practical path planning in fully dynamic environments. They have exploited the fuzzy modeling to define Fuzzy Artificial Potential Fields (FAPF) which provides a real-time and flexible path planning. It is shown that FAPF paves a way to merge both global and local path planning strategies. Simulations show that the planner is both very fast and capable of handling the local minima which can trap mobile robots before reaching the goal. Based on the literature survey on motion prediction models it is observed that i)The existing models lack flexibility in handling the uncertainties of the real-life situations; ii) Probabilistic models sometimes fail to model the real-life uncertainties; iii) The existing prediction techniques show poor response time due to their complex algorithmic structure; iv) Most of the approaches validate the results with simulated data or simple navigational environments.

The present work overcomes these difficulties with a novel solution for short term motion prediction using fuzzy rule-based prediction technique. History of moving object motion positions is captured in the form of fuzzy rule base, and the next instance object position is predicted using fuzzy inference process. Because of the multi-valued nature of fuzzy logic, this approach enjoys high robustness in dealing with noisy and uncertain data. However, direct implementation of the rule base is not suitable for real-life navigation systems due to the formation of huge number of rules. The total number of fuzzy rules to be used are directly proportional to the number of fuzzy sets defined for the application and the number of fuzzy members present in each fuzzy set. Inconsistent and redundant rules identified in the rule base are optimized by defining directional space within navigational space and decision tree approach.

The authors in their previous work [16] have implemented the extraction of objects of interest within the robotic navigational environment from the stereo vision system. Hence the focus of the present work is only limited to the prediction of the moving object's motion within the navigational environment.

The paper is organized as follows. In section 2, fuzzy rule-based object motion prediction process is explained. Sections 3 and 4 discuss the optimization of the fuzzy rule-base using directional space approach and decision tree approach.

In section 5 the fuzzy rule-base implementation details are presented. Experimental results are presented in section 6. Finally, concluding remarks are given in section 7.

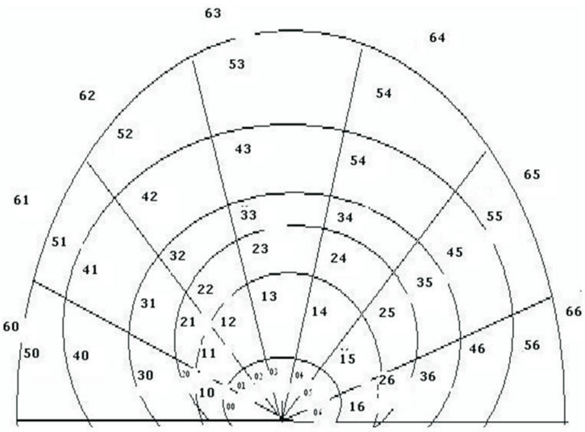
2. Fuzzy Rule-based Object motion prediction

The difficulty of dynamic obstacle motion prediction lies on the uncertainty of obstacle motions. In the proposed work we have considered intentional motion model for the moving objects within the navigational environment. Motion state of an obstacle at time t is generally represented by $(p(t),v(t),a(t))$ which represent the position, velocity and acceleration of the object at time t. In this model an obstacle moves in a scheduled route, such as a predetermined destination, or a programmed route. The obstacle may also try to avoid collision with others. In this case we have,

$$a(t) = a(t-dt) + \beta e(t) \tag{1}$$

Where $e(t)$ represents the variations in the accelerations resulting from any internal or external forces of the obstacle a and β are any two constants that specify the tendency of acceleration change. The function $e(t)$ depends on the particular environmental conditions. It differs from the random motion model in the way that, $e(t)$ cannot be described by any probability distribution. The acquisition of $e(t)$ relies very much on the background knowledge of the obstacles and a through observation of the history of motion of the moving objects. Fuzzy logic is an important branch of intelligent robotics. It does not need to establish accurate mathematical models and it is easy to construct its control structure with good robustness.

In the proposed work, the navigational environment is modeled as a fuzzy world model. The robot is capable of visualizing the navigation environment in front (about 180 degrees in semi circular range). Fuzzy regions in front of the robot are defined according to the visualization capability of the sensors. Each object detected has a distance variable from the Robot. This range data has a different membership in each of the 7 range subsets defined as Very very far (VVFAR), Very far (VFAR), Far (FAR), Moderate (MOD), Near (NEAR), Very near (VNEAR), Very very near (VVNEAR). The direction of universe is divided into 7 subsets. The linguistic variables that describe the angle heading are Very very left (VVLEFT), Very left (VLEFT), Left (LEFT), Front (FRONT), Right (RIGHT), Very right (VRIGHT), Very very right (VVRIGHT). The fuzzy representation of the environment is shown in Figure 1 with numerical notation for each region. The fuzzy representation divides the whole navigation environment into different regions like VVFAR-VLEFT (61), FAR-RIGHT(44) and NEAR-FRONT(23) etc.



a. Division of space in Fuzzy subsets of Range and Direction

Fuzzy Range	Numeric Notation	Fuzzy Direction	Numeric Notation
VVNEAR	0	VVLEFT	0
VNEAR	1	VLEFT	1
NEAR	2	LEFT	2
MOD	3	FRONT	3
FAR	4	RIGHT	4
VFAR	5	VRIGHT	5
VVFAR	6	VVRIGHT	6

b. Numeric notation for Fuzzy Range and Direction

Fig. 1. Division of Navigation Space into Fuzzy subsets of Range and Direction.

As the regions defined are fuzzy in nature, there can be overlaps from one region to another region. For simplicity these overlaps are not shown in the figure. The range and angle information need to be represented by a suitable membership function. Many authors have addressed critical issues relating to the selection and performance of fuzzy membership functions for various real-time robot control applications [6], [8], [14]. In most of the cases triangular membership function has proved superior over other membership functions like trapezoidal, Gaussian, bell shaped, polynomial-PI and sigmoidal. For our application as the prediction needs to be more accurate and the strength of the rule/ rules fired can make remarkable difference in the prediction, selection of triangular membership function for representing angle and range values is inevitable. The selection of 07 fuzzy subsets for range and angle is moderate as selecting 05 categories will have less number of fuzzy rules but, quality of prediction may reduce if navigation space is large, selecting 09 or more number of categories will increase the number of fuzzy rules as well as the complexity of the system which could reduce the response time of the predictor. Both range and angle subsets are normalized between 0-1.

In the rule base formation phase, rules are defined and added to the rule base using real-life data consisting of human motion patterns with velocity in the range 2-10 kmph. At time $t1$, the position (angle and range) of the moving object from the robot is read. Using fuzzification the observed data is converted to fuzzy value. At time $t2$ ($t2 > t1$ and $t2$ - t1 > δ, where δ is threshold time difference greater than or equal to 1 sec) the sensor reads the position of the same object. The reason for considering $\delta \geq 1$ sec is that, the time needed to process the captured image to identify the objects of interest by the vision system needs at least 01 sec or more as per the current literature. The maximum value of the δ considered was 04 seconds. This is because, as the time gap between the measurements increases the quality of the prediction reduces as well as the prediction looses its significance. The read value of the object position is converted to fuzzy value. The same process is followed at time $t3$ ($t3 > t2$ and $t3 - t2 = t2 - t1$) to get the fuzzy value of the location of the same object under observation. A fuzzy rule with the positions of the moving object at time $t1$ and $t2$ as the antecedent and the position of the object at time $t3$ as the consequent is formed and added to the rule-base. Each rule in the rule-base is represented as

$$\text{IF } (R1, \theta 1) \text{ and } (R2, \theta 2) \text{ THEN } (R3, \theta 3)$$

where $R1$ and θ1 represent the range and the angle respectively of the object at time $t1$, $R2$ and $\theta\,2$ represent the range and the angle respectively of the object at time $t2$, and $R3$ and $\theta\,3$ represent the range and the angle respectively of the object at time $t3$. Similar rules are added to the rule-base for different objects observed at various positions in the navigation environment.

In the implementation phase of the predictor, the robot observes the moving object at time $t1$ and $t2$ and sends the data to the fuzzy predictor algorithm. With the application of fuzzy inference process, prediction of the next instance position of the moving object is carried out. The complete process of short term motion prediction is represented in Figure 2.

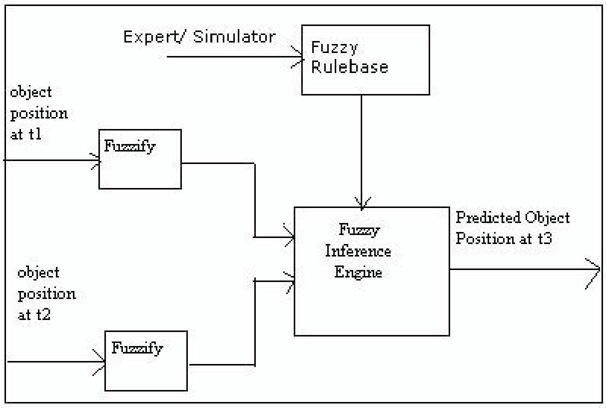


Fig. 2. Short term motion prediction.

3. Optimization of the Rulebase by Partitioning the Navigational Space

Many of the rules defined in the system look inconsistent such as

1. IF object at $t1$ is VFAR,VVLEFT and object at $t2$ is FAR,VVLEFT THEN object predicted at $t3$ is MOD,VVLEFT.
2. IF object at $t1$ is VFAR,VVLEFT and object at $t2$ is FAR,VVLEFT THEN object predicted at $t3$ is FAR,VVLEFT.

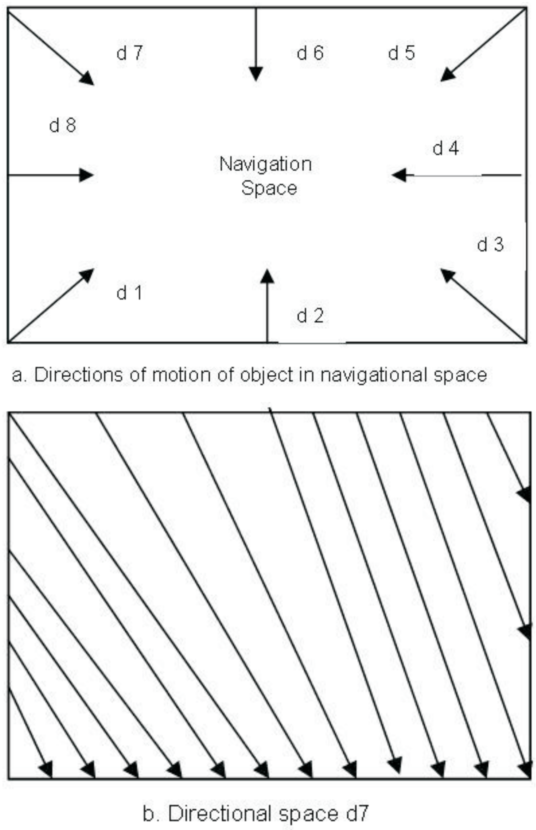


Fig. 3. Division of navigational space into Directional Space.

Where the antecedents are same but the consequents are different. The reason for the inconsistency is due to the direction of traversal of the object. Future motion of the object is dependent on the history of the direction of the traversal of the object. To overcome this type of inconsistency, while defining the rule base, partitioning of the navigational space is done. Considering the navigational space that is tessellated in eight geographical directions, the sensor readings of the object positions taken at previous two time intervals forms a trajectory in one of these directions.

$$\{SW(d1), S(d2), SE(d3), E(d4), NE(d5), N(d6), NW(d7), W(d8)\}$$

A separate directional space is created for each direction (Figure 3) and rules are clustered based on the direction of traversal object. Depending on the direction of traversal of the object, only those rules which belong to that directional space will be selected for processing.

4. Rulebase Optimization using Decision tree Approach

The proposed fuzzy predictor algorithm has to process all the rules in a sequential form. The time complexity of the algorithm is linear and is of the order $O(n)$. This is reduced by reordering the rules in the form of a decision tree. Each group of rules in the directional space is reorganized and IF-ELSE statements are written in the form of a decision tree. The decision tree is a classifier in the form of a tree structure, where each node is either a leaf node - indicates the value of the target attribute (class) of examples or a decision node specifies some test to be carried out on a single attribute-value, with one branch and sub-tree for each possible outcome of the test.

Considering the basic organization of the fuzzy rulebase (which is a sequential set of rules) for two rules

Rule1: IF $((R1==2, \theta1 == 2))$ and
IF $((R2==1, \theta2 == 1))$ THEN $R3, \theta3 = 21;$

Rule2: IF $((R1==2, \theta1 == 2))$ and
IF $((R2==1, \theta2 == 2))$ THEN $R3, \theta3 = 22;$

We can have rules

i) starting with $R1=2$ and $\theta1$, $R2$, and $\theta2$ with any values from 0-6
ii) starting with $R1=2$ and $\theta1 = 2$ and $R2$ and $\theta2$ with any values from 0-6
iii) starting with $R1=2$, $\theta1 = 2$, $R2=1$ and $\theta2$ with any values from 0-6
iv) starting with $R1=2$, $\theta1 = 2$, $R2=1$, and $\theta2$ with any values from 0-6

These set of rules when organized in sequential order form a huge number of rules and consequently increasing the size of the rule base for processing. Using decision tree approach the two rules defined previously can be reorganized as follows.

```
if(R1==2)
{
    if(θ1 == 2)
        {
            if(R2==1)
                {
                    if( θ2 == 1)
                {R3, θ3 = 21;}
                    if(θ 2 == 2)
                {R3, θ3 = 22; }
                    }
        }
    }
```

In the above expression if $R1 \neq 2$, no expression within the if block of $R1$ is executed. Similarly all the rules in the fuzzy rulebase can be reorganized in the form of a decision tree. For the developed rule-base, Figure 4 gives the partial representation of the decision tree for IF-ELSE statements. The input read by the fuzzy predictor algorithm classifies the input set to one of the directional spaces (1 to 8) defined in Section 3. Each internal node v is labeled with an integer 1 to 8 indicating the direction of traversal of the moving object and one of the directions will be selected based on the history of object motion. Each level in the decision tree corresponds to a fuzzy set indicating either the range or direction subsets $(R1, \theta1, R2, \theta2)$. Each item in the fuzzy antecedent is processed as and when it receives inputs at each level in the decision tree and each input is a partial information of the position of the object in the navigational environment. Each interior node in the decision tree corresponds to a variable; an arc to a child represents a possible value of that variable. A leaf represents a possible value of target variable given the values of the variables represented by the path from the root. Based on the input one of the outgoing edges will be selected. The outgoing thick edge represents the selected fuzzy subset and remaining dotted edges represent the other unselected nodes.

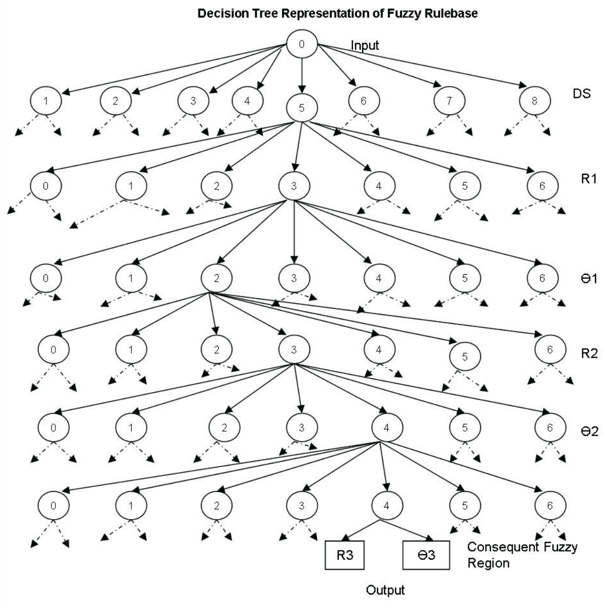


Fig. 4. Optimization of Fuzzy rule-base using Decision tree.

To execute the algorithm, the process starts at the root node v, follows the edge labeled $f(nv)$, and continues recursively. Thus, the execution of the algorithm gives a path from the root to some leaf. Each leaf has an integer label; when the execution reaches a leaf, its label is returned as the algorithm's output.

Let $T(A, f)$ be the length of the root-to-leaf path in decision tree A traversed when the input is f(rule). The complexity $T(A)$ of any decision tree algorithm A is its depth and the complexity of the problem is the depth of the shallowest decision tree. For the prediction algorithm, the time complexity of the decision tree representation of the rule-base system is given by

$$T(n) = O(n) \quad (2)$$

where n is the depth of the tree.

Table 1 represents the selection of the nodes of the decision tree at various levels.

Table 1. Decision Tree Analysis for Short Term Motion Prediction.

Level	Nodes	Type
0	DS={1,2,3,4,5,6,7,8}	Directional Space
1	IF (R1 and θ1) and (R2 and θ2)	Rule Antecedent
2	R1={0,1,2,3,4,5,6}	Fuzzy Object distance at time t1
3	θ1={0,1,2,3,4,5,6}	Fuzzy Object angle at time t1
4	R2={0,1,2,3,4,5,6}	Fuzzy Object distance at time t2
5	θ2={0,1,2,3,4,5,6}	Fuzzy Object angle at time t2
6	R3={0,1,2,3,4,5,6} and θ3={0,1,2,3,4,5,6}	Rule Consequent: Predicted Fuzzy Region at time t3

5. Fuzzy Rule-base Implementation for Motion Prediction

The rule-base implementation comprises of the observation of the moving objects in the navigational environment at equal time intervals and prediction of their future position using the Fuzzy rule-base. This step involves the fuzzy inference process. The Fuzzy inference process comprises five parts: fuzzification of the input data, application of the fuzzy operator (AND or OR) in the antecedent, implication from the antecedent to the consequent, aggregation of the consequents across the rules and defuzzification. The fuzzy inference process adopted the Mamdani model. The Mamdani model uses rules whose consequent part is a fuzzy set.

$$Ri\text{: if } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ and } x_3 \text{ is } A_{is} \text{ then } Y \text{ is } C_i \quad i = 1,2,3 \ldots M \qquad (3)$$

where M is the number of fuzzy rules, $x_j \in U_j$ ($j = 1, 2, 3 \ldots s$) are the input variables, $y \in V$ is the output variable and $_{Aij}$ and C_i are fuzzy sets characterized by membership functions $\mu_{Aij}(x_{ij})$ and $\mu_c{}'_i$, respectively.

Given the inputs of the form

$$x1 \text{ is } A'1, x2 \text{ is } A'2, \ldots X_r \text{ is } A'_r$$

where $A'_1, A'_2, \ldots A'_r$ are Fuzzy subsets of $U_1, U_2, \ldots U_r$. The contribution of rule R_i to Mamdani model's output is a Fuzzy set whose fuzzy membership function is computed by

$$\mu_c{}'_i(y) = \mu_{A'1}(x_1)^{\mu}{}_{A'2}(x_2) \ldots \mu_{A'j}(x_j)\} \qquad (4)$$

where ^ denotes the 'min' operator. The final output of the model is the aggregation of outputs from all the rules using the max operator.

$$\mu_C(y) = max\{\mu_C{}'1(y), \mu_C{}'2(y), \ldots \mu_C{}'_L(y)\} \qquad (5)$$

Defuzzification of the final output is done to get the crisp value. Three most commonly used defuzzification techniques are considered: i) Fuzzy OR method/Min-Max, ii) Center Of Area (COA) and iii) Mean Of Maximum (MOM) methods. These methods operate on range and angle output subsets separately to generate the final crisp value, indicating the range and angle of the final output.

6. Experimental Results

Table 2. Evolution of Short term predictor.

Development Stage	Number of Rules to be processed in the Worst case	Average Time Complexity
Basic Unoptimizeed Fuzzy predictor	1200	$O(n)$ where n is the number of Fuzzy Rules
Predictor with Directional Space Approach	140 (Approx)	$O(n')$ where n' is the number of Fuzzy Rules and $n' < n$
Predictor with Decision Tree Approach	43(Approx)	$O(log\, n')$

Table 2 represents the evolution of the Fuzzy predictor algorithm. The table is parameterized by the stage of the algorithm development, the number of rules to be processed and the time complexity. The unoptimized fuzzy predictor consists of all the rules identified during the formation of the rule-base. As the rule-base is large and consists of inconsistent rules, its response time and relative error is high. All the rules are processed in a linear order, the time complexity of the predictor is $O(n)$ where n is the number of rules. The directional space approach clusters the rules in different directions which reduces inconsistency, as well as response time. The decision tree approach reorganizes the rule-base and reduces the response time and time complexity of the predictor to $O(logn')$ where n' is the number of rules processed by the predictor algorithm. The fuzzy predictor algorithm is developed in C++ language.

The algorithm is tested on 1.66 GHz machine in VC++ environment. The tests are carried out for real-life benchmarked datasets[3], [11], [13]. These data sets are gathered through i) INRIA Labs with data captured at INRIA Labs at Grenoble, France (A wide angle camera lens in the entrance lobby of the INRIA Labs at Grenoble, France. The resolution is half-resolution PAL standard); ii) Motion Capture Web group of Univ. of S. California (Consisting of Human Motion Patterns); iii) CMU Graphics Lab dataset. (Vicon motion capture system consisting of MX-40 cameras with images of 4 megapixel resolution).

The data sets consist of different human motion patterns. These include people walking alone, running, meeting with others, window shopping, entering and exiting shops (average speed in the range 2-10 kmph). The position of the moving objects within the navigational environment at any instant of time is given separately as a database so that any prediction algorithm can be tested and analyzed for any number of objects. These motions exhibit intentional motion and predicting the next instance position of objects in such scenario is an important task as it can find applications in keeping track of human motion patterns in hospitals, shopping complex and in exhibition halls etc.

Figure 5 represent the movement of the objects from left to right direction and the corresponding short term motion prediction path. Pi and Ai represent the predicted and the actual path traversed by the moving object. $Pi(G)$ and $Ai(G)$ represent the predicted goal and the actual goal of the object. $A1$ is the actual path observed and $A1(G)$ is the actual goal reached by the object $A1$.

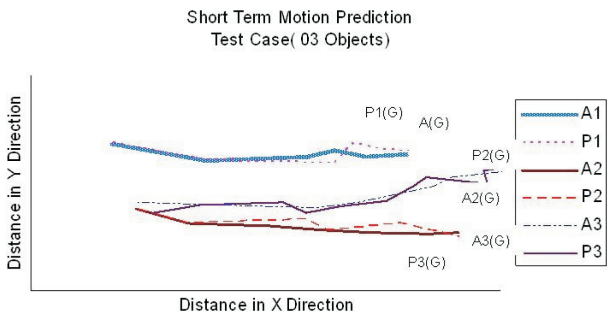


Fig. 5. Prediction graphs showing few of the path prediction solutions for Short term motion prediction.

We define the Relative Error (RE) for M sample test data (sum of the number of predicted positions for a specific object in motion) as

$$RE = \frac{(da - dp)}{da}/M \qquad (6)$$

Where da is the actual position, dp is the predicted position of the moving object in the navigational environment.

The average relative error is calculated for various test cases using Min Max, MOM and COA defuzzification techniques. For each test case the average response time is also calculated to find its suitability to real-life environment. For measuring the performance of the system the standard parameters like prediction steps and relative error are used. The prediction algorithm is tested with prediction steps 02 seconds (Fig. 6), 03 seconds (Fig. 7), 04 seconds (Fig. 8).

Table 3 represents the results of the Short term predictor at various stages of development. Each stage in the evolution of the fuzzy predictor is parameterized by the relative error and average response time. These prediction steps indicate the in between time gap for each successive measurement (of the object position) by the vision system.

Variations in the velocity and directions of motion of the moving objects in these test cases are the sources of uncertainty in predicting the next instance position of the moving object. Tests are carried out to measure the relative error between the actual and predicted positions when minute variations in velocity and directions of the moving objects are observed (Fig. 9).

The proposed predictor generates the next instance position as a fuzzy region than as a (x,y) coordinate. This helps in the robot to classify the predicted region as a danger zone or the region of interest.

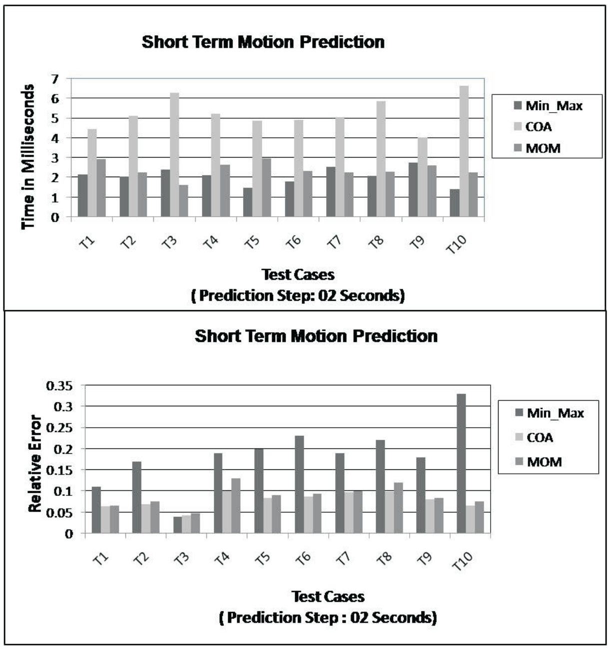


Fig. 6. Average response time and relative error of the Short term predictor at prediction step: 02 seconds.

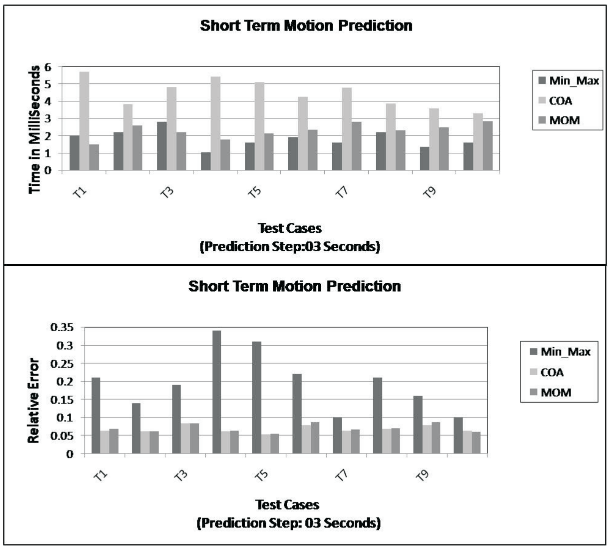


Fig. 7. Average response time and relative error of the Short term predictor at prediction step: 03 seconds.

Defuzzification of the output generates the predicted coordinate position of the moving object. The response time of the algorithm with Min Max defuzzification varied in the band from 1.45 milliseconds to 2.9 milliseconds and the relative error in the band from 0.04 to 0.4.

The response time of the algorithm with COA defuzzification varied in the band from 3 milliseconds to 7 milliseconds and the relative error in the band from 0.01 to 0.1. The response time of the algorithm with MOM defuzzification varied in the band from 1.95 milliseconds to 3.37 milliseconds and the relative error in the band from 0.04 to 0.1. From the graphs it is observed that the predictor with MOM defuzzification performs better in terms of response time with less relative error.

Table 3. Results of Short term predictor at various stages of development.

Development Stage	Relative Error	Average Response time in millisec
Basic Unoptimized Fuzzy predictor	1-20%	500
Predictor with Directional Space Approach	1-15%	15-20
Predictor with Decision Tree Approach	1-10%	2-5

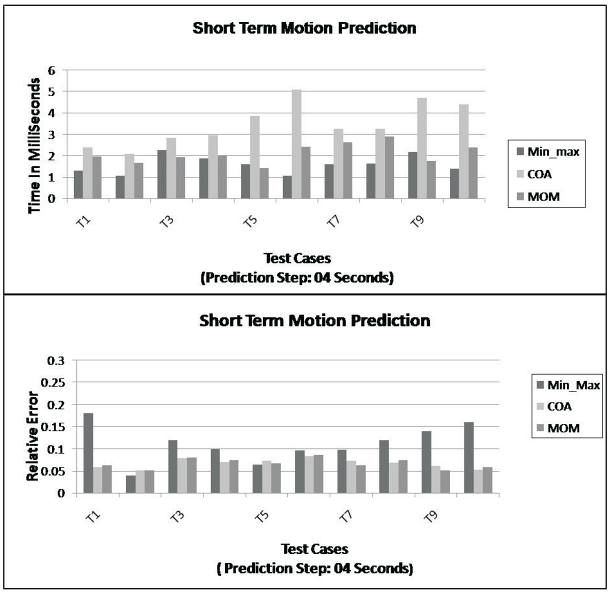


Fig. 8. Average response time and relative error of the Short term predictor at prediction step: 04 seconds.

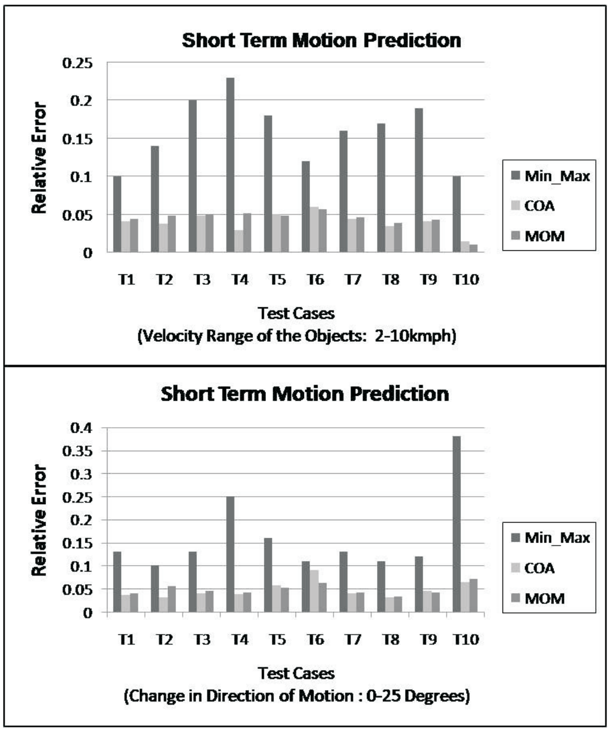


Fig. 9. Relative Error when measured with variations in velocity and direction of motion of moving objects.

Table 4. Comparison of Short term predictors.

Short Term Predictor	Relative Error	Response time in seconds
Neural Network predictor[8]	6-17%	560x 10^{-3} sec
Bayesian Occupancy Filters[10]	1-10%	100x 10^{-3} sec
Extended Kalman Filter[12]	1-20%	0.1 sec
Fuzzy Predictor with MOM	1-10%	02x 10 sec to 05x 10^{-3} sec

A few of the well known motion prediction techniques are re-implemented and are compared with the developed fuzzy predictor in respect of response time and relative error (Table 4). From the table it can be observed that the performance of the predictor is comparable with regard to relative error but better than the other prediction methods as far as response time is concerned.

7. Conclusion

In a dynamic navigation system the robot has to avoid stationary and moving objects to reach the final destination. Short term motion prediction for moving objects in such an environment is a challenging problem. This paper proposes a simplified approach for predicting the future position of a moving object (human motion patterns) using fuzzy inference rules derived from experts knowledge and real-life data. The rule-base has been optimized by directional space approach and decision tree approach. Fuzzy based prediction is more flexible, can have more real life parameters, comparable to the existing approaches and suited for real-life situations. The results of the study indicate that, the fuzzy predictor algorithm gives comparable accuracy with quick response time when compared to existing techniques.

ACKNOWLEDGMENTS

The authors are thankful to the benchmark dataset provided by EC Funded CAVIAR project, CMU Graphics lab and Motion capture web group. We are indebted to AICTE, Government of India, for funding our project.

AUTHORS

Vijay S. Rajpurohit* - Department of Computer Science and Engineering, Gogte Institute of Technology, Belgaum, 590008, India. E-mail: vijaysr2k@yahoo.com.
M.M. Manohara Pai - Department of Information and Communication Technology, Manipal Institute of Technology, Manipal, 576104,India.
E-mail: mmm.pai@manipal.edu.
* Corresponding author

References

[1] Foka Amalia, Trahanias Panos E., "Predictive Autonomous Robot navigation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, EFPL, Lausanne, Switzerland, Oct. 2002, pp. 490-494.
[2] Foka Amalia, Trahanias Panos E., "Predictive control of

Robot velocity to avoid objects in dynamic environments". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 27th-31st October, 2003, Las Vegas, Nevada, pp. 370-375.
[3] Gardner Andrew, Pighin Fred, 2001, *Motion Capture Web*, Retrieved July 2008, from University of Southern California websitehttp://projects.ict.usc.edu/graphics/animWeb/humanoid/.
[4] Chang Charles C., Song Kai-Taj, "Dynamic motion planning based on Real-Time object Prediction". In: *Proceedings of the 1996 IEEE Intl. Conference on Robotics and Automation*, Minnesota, USA.
[5] Laugier C., Petti S., "Navigation in Open and Dynamic Environments". In*: IEEE International Conference on Robotics and Automation (ICRA'05)*, 2005, Barcelona, Spain.
[6] Huang Hong, Chang Liuchen, "Simulation study on the membership functions of the basic logic controller for the electric vehicle propulsion systems". In: *Canadian Conference on Electrical and Computer Engineering,* Calgary, Canada, May 1996, pp. 1004-1007.
[7] Yu Huiming, Su Tong, "A Destination Driven Navigator with Dynamic Object Motion Prediction". In: *International Conference on Robotics 2L Automation,* Seoul, Korea. 21st-26th May, 2001, pp. 2692-2697.
[8] Zhao Jin, Bose Bimal K., "Evaluation of membership functions for fuzzy logic controlled induction motor drive". *IECON Proceedings (Industrial Electronics Conference),* volume 1, 2002, pp. 229-234.
[9] Kalman R., "A New Approach to Linear Filtering and Prediction Problem", *Transactions of the ASME-Journal of Basic Engineering,* vol. 82, 1960, pp. 35-45.
[10] Webb P., Fayad C., Brettenbatch C., "The integration of an optimized Fuzzy logic navigation algorithm into a semi-autonomous Robot Control System". In: *International Workshop on Recent Advances in Mobile Robots*, Leicester, UK, 29th June 2000.
[11] *Motion Capture Database*, Retrieved July 2008 from Carnegie Mellon University Motion Capture Database website: http://mocap.cs.cmu.edu/, funding from NSFEIA-0196217.
[12] Madhavan R., Schlenoff C., *Moving Object Prediction for Off-road Autonomous Navigation*. SPIE Aerosense conference, Orlando, 21st-25th April, 2003.
[13] Fisher Robert, Santos-Victor Jose, James Crowley (2001), *CAVIAR Video Sequence Ground Truth*, Retrieved Jan 2008,fromhttp://homepages.inf.ed.ac.uk/rbf/CAVIAR/, funded by EC Funded CAVIAR project/IST 2001 37540.
[14] Berkan Riza C., Trubatch Sheldon L., *Fuzzy Systems Design Principles: Building Fuzzy If-Then Rule Bases*. John Wiley & Sons Inc., 1997.
[15] Irajit R., Manzuri-Shalmanit M. T., "A New Fuzzy-Based Spatial Model for Robot Navigation among Dynamic Obstacles". In: *IEEE International Conference on Control and Automation*, Guangzhou, China, 30th May - 1st une, 2007, pp. 1323-1328.
[16] Rajpurohit Vijay S., Pai Manohara M. M., "Feature Extraction Learning For Stere*ovision Based Robot* Navigation System". In: *14th IEEE International Conference on Advanced Computing Communications - ADCOM 2006*, pp. 362-365, ISBN-1-4244-0715-X, NITK, Surathkal, Mangalore, 20th-23rd, December 2006.
[17] Nam Yun Seok, Lee Bum Hee, Kim Moon Sang, "View-Time Based Moving Object Avoidance using Stochastic Prediction of Object Motion". In: *IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, USA, April 1996.
[18] Hui-Zhong Zhuang, Du Shu-Xin, Wu Tie-jun, "On-line real-time path planning of mobile Robots in dynamic uncertain environment". *Journal of Zheing University Science A* , 2006, pp. 516-524.
[19] Zhu Q., "Hidden Markov Model for Dynamic Object Avoidance of Mobile Robot Navigation", *IEEE Transactions on Robotics and Automation*, 1991, pp. 390-396.