

Bayesian Inference Driven Behavior Network Architecture for Avoiding Moving Obstacles*

Hyeun-Jeong Min and Sung-Bae Cho

Dept. of Computer Science, Yonsei University
134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, Korea
{solusea, sbcho}@cs.yonsei.ac.kr

Abstract. This paper presents a technique for an intelligent robot to adaptively behave in unforeseen and dynamic circumstances. Since the traditional methods utilized the relatively reliable information about the environment to control intelligent robots, they were robust but could not behave adaptively in complex and dynamic world. On the contrary, behavior-based approach is suitable for generating autonomous behaviors in the environment, but it still lacks of the capabilities to infer dynamic situations for high-level behaviors. This paper proposes a 2-layer control architecture to generate adaptive behaviors, which perceive and avoid dynamic moving obstacles as well as static obstacles. The first level is to generate reflexive and autonomous behaviors with the behavior network, and the second level is to infer dynamic situation of mobile robots with Bayesian network. Experimental results with various situations have shown that the robot reaches the goal points while avoiding static or moving obstacles with the proposed architecture.

1 Introduction

It is difficult to avoid moving obstacles because a mobile robot has to perceive situations by only utilizing its sensors in real-time. The predictability is central to the collision avoidance in non-stationary conditions. If there is no need to predict, we can rely entirely on what is sensed, resulting in a purely reactive approach [1]. However, it is difficult to predict in the real world, even in the best of circumstances. In MIT AI Lab., they proposed a method of learning each circumstance for avoiding moving obstacles [2]. Hashimoto utilized the evolutionary computation and fuzzy system for avoiding moving obstacles [3], and Inoue presented a behavior learning method based on Bayesian networks and experience of interaction between human and robots of avoiding moving obstacles [4]. Nicolescu and Mataric dealt with changing situations by learning and constructing hierarchical structure of previous behaviors for solving the problem of avoiding moving obstacles [5].

The behavior network proposed by Maes [6] can acquire global goals as well as autonomously select behaviors as bestowing goals on the behavior-based system. Although behavior network is proposed for solving problems with goals in uncertain environments, it is insufficient to generate adaptive behaviors in changing and complex situations. To cope with this problem we propose a control architecture in which

* This research was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D programs funded by the Ministry of Commerce, Industry and Energy of Korea

Bayesian network controls behavior network. This control architecture selects a behavior of the highest weight which is computed by the inference of Bayesian network in behavior network. Bayesian network has advantage that it is independently applied to environment because it is designed by only using the information from sensors.

2 Bayesian Inference Driven Behavior Network

The variables required for the mathematical notations of the proposed method for Bayesian inference driven behavior network are as follows. These variables are also applied to the function for the Bayesian inference [7], the computation of an activation level of each node, and the selection of an active node in the behavior modules.

- α : Activation level
- θ : The initial value of the global threshold which is reduced by 10% if no executable node has an activation greater than it
- ϕ : The weight of environmental sensor inputs and successor links
- γ : The weight of goal inputs and predecessor links
- δ : The weight of protected goal inputs and conflictor links
- t : Current time
- B : A set of behavior nodes
- B_{PS} : A set of preconditions of behavior node b (see table 1)
- B_{AS} : A set of add lists of behavior node b (see table 1)
- B_{DS} : A set of delete list of behavior node b (see table 1)

The function for selecting the activation node b_i in time t is defined as follows.

$$b_i(t) = \begin{cases} 1 & \text{if } \begin{cases} \alpha_{b_i}(t) \geq \theta \\ executable(b_i, t) = 1 \\ \alpha_{b_i}(t) \geq \alpha_{b_j}(t), \forall j \in \cdot(1) \text{ and } (2) \end{cases} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$executable(b_i, t) = \begin{cases} 1 & \text{if } b_i \text{ is executable at time } t \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The function for the activation level α_{b_i} in behavior node b_i is defined as follows.

$$\alpha_{b_i}(t) = \begin{cases} 0 & \text{if } t = 0 \\ D(b_i, t) + \beta(b_i, t) & \text{otherwise} \end{cases}$$

From this function, the activation value of each behavior $D(b_i, t)$ is defined as follows.

$$D(b_i, t) = S(b_i, t) + G(b_i, t) - P(b_i, t) + \sum_{b_j, b_k} (SP_{EW}(b_j, b_k, t) + SP_{FW}(b_j, b_k, t) - SP_{PG}(b_j, b_k, t)) \quad (3)$$

$$SP_{EW}(b_j, b_k, t) = \begin{cases} 1 & \text{if } \sigma \in B_{PS}, \sigma \in B_{AS} \\ 0 & \text{otherwise} \end{cases}$$

$$SP_{FW}(b_j, b_k, t) = \begin{cases} 1 & \text{if } \sigma \in B_{AS}, \sigma \in B_{PS} \\ 0 & \text{otherwise} \end{cases}$$

$$SP_{PG}(b_j, b_k, t) = \begin{cases} 1 & \text{if } \sigma \in B_{DS}, \sigma \in B_{PS} \\ 0 & \text{otherwise} \end{cases}$$

In the above activation function $\square(b_i, t)$ is the weight affecting the activation in behavior nodes after the agent infers situation by Bayesian network, and it is defined as follows.

$$\beta(b_i, t) = \begin{cases} \xi & \text{if } b_i \in r_{k,i}, i \in I, \Psi(k, i) = \sigma \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$r_{k,i} = \{s \mid s \in \text{effect_node}(k), s : \text{ith state}\}$$

$$I = \{x \mid x \in S_j(t), 1 \leq x \leq \#(\text{Effect Nodes})\} S_j(t) : \text{cause at time } t$$

$$\Psi(k, i) = P(b_{ki} \mid c_{1j} \cdots c_{mj}) : \text{conditional probability, } m : \#(\text{cause nodes})$$

$$\sigma = \text{Max}\{\Psi(k, i) \mid 1 \leq i \leq \#(\text{states in } r_i), k : \text{effect node}\}$$

In equation (3), $S(b_i, t)$ is the activation value from sensors at time t , $G(b_i, t)$ is the activation value from goals at time t , and $P(b_i, t)$ is the activation value to be deleted from protected goal at time t . In addition to these equations, $SP_{EW}(b_j, b_k, t)$ is the backward spreading from node b_j to b_k and it means both the preconditions of b_j and the add list of b_k and $SP_{FW}(b_j, b_k, t)$ is the forward spreading from node b_j to b_k , and it means both the add list of b_j , and the precondition of b_k . $SP_{PG}(b_j, b_k, t)$ is to take away from node b_j to b_k and it means both the delete list of b_j and the precondition of b_k .

Bayesian network represents all the events with DAG (Directed Acyclic Graph) and each node in DAG models probabilistic independency. It consists of the relation of cause and effect nodes from their probabilities and can infer the results from conditional probability of cause nodes. In Bayesian network, each node corresponds to probabilistic variables C_i , C_j , and E , which are the causes of the effect node and each link is associated with conditional probabilities between linked variables.

In equation (4), $r_{k,i}$ means the result node of Bayesian network at the i th state in the k th effect, and I is the set of indexes of result nodes where preconditions are satisfied at time t . $\Psi(k, i)$ is the conditional probability at the i th state in the k th result node, and σ is the state of the highest probability in the k th result node.

$$\text{IF } b_i(t) = 1 \text{ THEN execute } b_j \text{ ELSE } \theta = \theta \times 0.9 \text{ and repeat the procedure} \quad (5)$$

Table 1. Internal links and external links

Internal Link	
Predecessor Link	$(\rho = \text{false}) \wedge (\rho \in \text{preconditions of node A}) \wedge (\rho \in \text{add lists of B})$
Successor Link	$(\rho = \text{false}) \wedge (\rho \in \text{add lists of node A}) \wedge (\text{A is executable}) \wedge (\rho \in \text{preconditions of B})$
Conflictor Link	$(\rho = \text{true}) \wedge (\rho \in \text{preconditions of node A}) \wedge (\rho \in \text{delete lists of B})$
External Link	
Sensors	$(\rho = \text{true}) \wedge (\rho \in \text{preconditions of node A})$
Goals	$(\rho \geq 0) \wedge (\rho \in \text{add lists of node A})$

3 Experimental Results

For experiments, we use the YAKS which is the 3D robot simulator, and the experimental environment is shown as Figure 1 (a). The environment of this simulator has

the mobile robot (number 0) which generates behaviors using the proposed method and two robots (number 1 and 2) which act as the moving obstacles. These two robots like obstacles can only detect and avoid pens, and cannot avoid a robot when they collide with other robots. This simulator may change the angle of the view within this simulator. In Figure 1 (a), the white cylinders represent the static obstacles, and they have changeable count in the experiments. The robot randomly starts from any positions for the comparative analysis when using the only behavior network and the proposed method, and also starts from the static position for the various analyses of the proposed method; we compared the success rates to avoid moving obstacles using the only behavior network and the proposed method respectively, and we investigated an avoiding direction in each case when using the proposed method.

The goals in these experiments are to avoid the moving obstacles and to reach the goal position marked with the light area in the corner of pens. We put randomly 5% of noise in the value from sensors because we may not predict the situation accurately in real world.

3.1 Behavior Network and Bayesian Network

Figure 1 (b) shows the behavior network architecture used in this paper, which has primitive behaviors like ‘Follow Light’, ‘Go Straight’, ‘Turn Left’, and ‘Turn Right.’ In the network, the predecessor links of ‘Follow Light’ are ‘Go Straight’ and ‘Near Light’, and the successor link is ‘Reach Goal’. The predecessor links of ‘Go Straight’ are also ‘Nothing’, ‘Turn Left’, and ‘Turn Right’, and the successor links are ‘Reach Goal’, ‘Avoid Obstacle’, and ‘Follow Light’. The predecessor links of ‘Turn Right’ are ‘Shade Area’, ‘Near Obstacle’, and ‘Go Straight’, and the successor links are ‘Avoid Obstacle’ and ‘Go straight’.

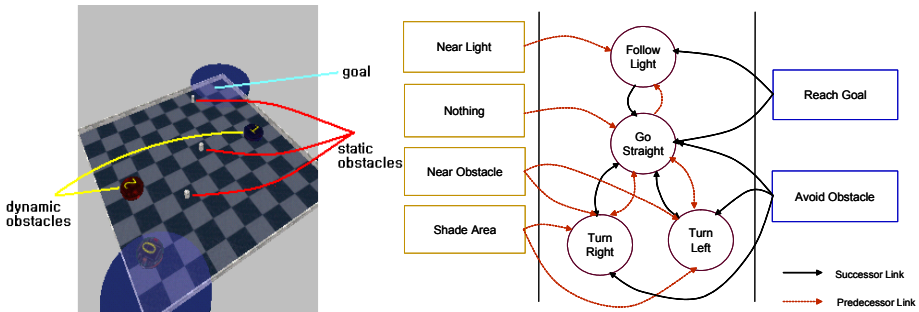


Fig. 1. (a) Experimental environment (left) (b) Behavior network (right)

Bayesian network designed for avoiding moving obstacles as well as static obstacles is as follows. The cause nodes from sensors are ‘distance 0’, ‘distance 1’, ‘distance 2’, ‘distance 3’, ‘distance 4’, ‘distance 5’, ‘distance 6’, and ‘distance 7’, and the cause nodes to infer the changing situations from previous behaviors are ‘distance of obstacles’, ‘position of obstacles’, ‘previous behavior’, and the directional change of obstacles like ‘rear_object’, ‘front_object’, ‘left_object’, and ‘right_object’. The initial probabilities of each node are normalized as $1/n$ for the n conditions in each node. According to this, the initial probabilities of ‘distance 0’ are $P(Near0)=0.5$ and

$P(Far0)=0.5$. There are some important factors to the nodes named ‘distance of obstacles’, ‘position of obstacles’, ‘previous behavior’, ‘left_object’, ‘right_object’, ‘front_object’, and ‘rear_object’ to infer dynamic situations.

For example, if left sensors of robot detect the obstacle in current state, the probabilities of $P(Near0)$ and $P(Near1)$ in the nodes ‘distance 0’ and ‘distance 1’ get higher. The probabilities of $P(Approach)$ and $P(GoAway)$ in the node ‘distance of obstacles’ and $P(Front2Left)$, $P(Left)$, and $P(Rear2Left)$ in the node ‘left_object’ are decided from the previous information doing by a robot. If $P(Approach) > P(GoAway)$ in the node ‘distance of obstacles’, the highest probability in those of $P(NoTurn)$, $P(TurnLeft)$, and $P(Rear2Left)$ in the node ‘left_turn’ is selected. The node ‘left_object’ consists of $P(Front2Left)$, $P(Left)$, and $P(Rear2Left)$ which mean the change of obstacle from front to left, on left, and from rear to left, respectively. The conditional probabilities of the effect node ‘left_turn’ are determined with the above probabilities of cause nodes.

3.2 Comparison

After the experiments to avoid moving obstacles using only behavior network and the proposed method, we have observed that a robot may not avoid moving obstacles when a moving obstacle comes from the opposite direction to a robot. Since a mobile robot avoids collisions as the direction and position of an obstacle from the robot, ‘Turn Left’ or ‘Turn Right’ is an important behavior. When starting at the same position and direction in the only behavior network and the proposed method respectively, the robot collides with moving obstacles in the only behavior network, but avoids the obstacle and reaches the goal in the proposed method.

We have analyzed the behavior for avoiding moving obstacles which get nearer from various positions and directions using the behavior network and the proposed method. We have verified the success of 52% and 90% in those of 60 trials using the behavior network and the proposed method, respectively. The cases of failure in the proposed method are when the obstacle gets nearer on the front side or changes its direction abruptly. Figure 2 shows the success rates for avoiding moving obstacles in the only behavior network and the proposed method. In this figure, y-axis is the success rate that ranges from 0 to 1.

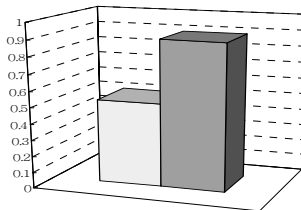


Fig. 2. The success rates for avoiding moving obstacles (Y-axis: Success rate, Dark bar: proposed)

Figure 3 shows the comparison of the behavior network with the proposed method. (a) is the comparison of the behavior sequences and (b) is that of the angles followed by the behaviors. In this figure, the solid line represents the behaviors and angles in

the behavior network and the dashed line represents the behaviors and angles in the proposed method. In the y-axis, 1 is ‘Go Straight’, 2 is ‘Turn Left’, and 3 is ‘Turn Right’. In the figure, the robot selects ‘Turn Left’ or ‘Turn Right’ when using the proposed method, but ‘Turn Right’ when using the only behavior network.

3.3 Analysis of Results

As mentioned before, we have analyzed the behavior selection of the robot at time t from the moved angles between the robot and obstacles using the proposed method. Figure 4 shows that the robot reaches the goal while avoiding moving obstacles, pens, and static obstacles of 2, 3, and 2 times, respectively. In this figure, (a) shows the trajectories of the robot and moving obstacles, and (b) shows the behavior selection of the robot to reach the goal and has the marked circles representing the collision avoidances with moving obstacles, static obstacles, and pens. In (a), blue line represents the trajectory of the mobile robot and black and red line represent moving obstacles, respectively.

We have subsequently analyzed behaviors of the robot in cases of the same direction and the different direction for comparing behavior selection as the changes of the angles between the robot and obstacles. Figure 5 shows the result of avoiding moving obstacles. In this figure, (a) and (b) are the trajectories in the cases of colliding with an obstacle 2 in the same direction on the left and on the right of the robot, and (c) represents the behavior sequences in (a) with respect to time. Lastly, (d) represents the behavior sequences in (a) with respect to time. In (c) and (d), x-axis is time line and y-axis is the selected behavior with respect to time. The numbers of y-axis represent that 1 is ‘Go Straight’, 2 is ‘Turn Left’, and 3 is ‘Turn Right’. Figure 6 shows the results to avoid moving obstacle with a real Khepera II mobile robot. In this figure, each of (a, c) and (b, d) shows the situations before and after avoiding a white box in front of the mobile robot.

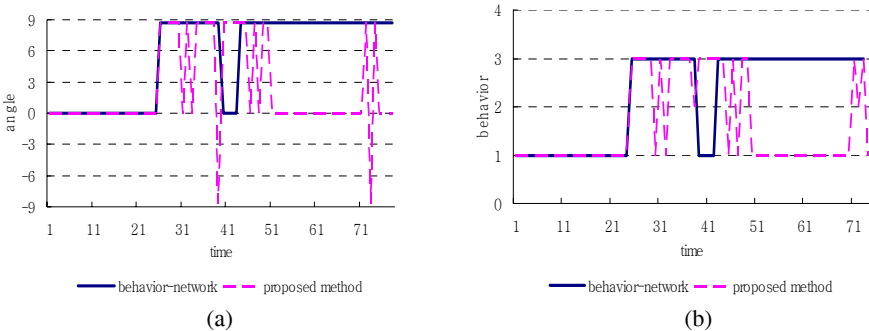


Fig. 3. The behaviors (a) and angles (b) in the behavior network and the proposed method

4 Conclusions

The experiments have validated that mobile robot goes to the goal with avoidance behaviors on moving obstacles using the Bayesian inference driven behavior net-

work. Autonomous behaviors of a mobile robot are generated by the inter-relations of sensors, goals and behaviors in behavior network, and adaptive behaviors of a mobile robot are also generated by expanding the behavior network through the inference of each situation using Bayesian network. The proposed method does not have limitation of re-constructing the system as the changes of environment like the systems with hybrid learning or planning, and overcomes the limitation to re-define many rules such as the systems using fuzzy rules and fuzzy inference [2].

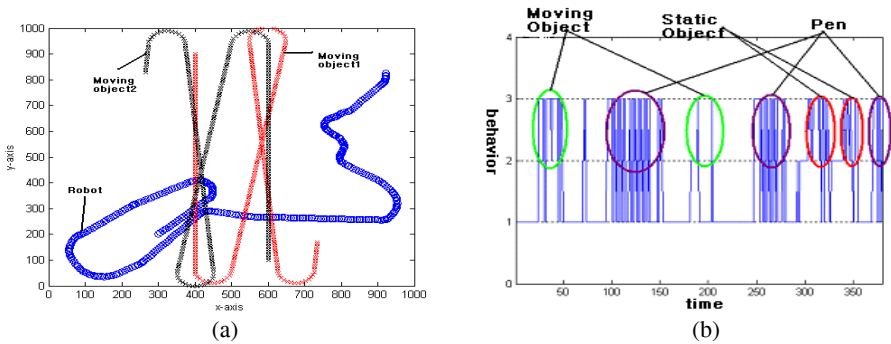


Fig. 4. The trajectory (a) and each behavior (b) by the time of avoiding moving obstacles in the proposed method

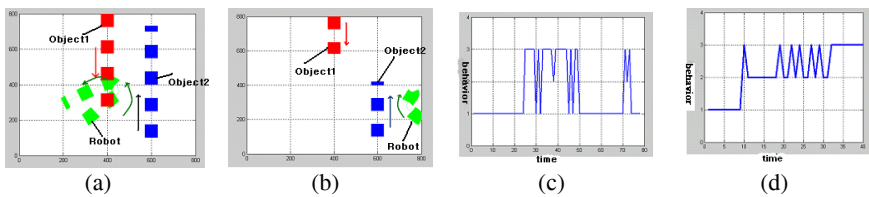


Fig. 5. Avoiding moving obstacles as the same direction with the robot using the proposed method. Each figure shows the avoidance in the right side of robot (a, c) and in the left side of robot (b, d)

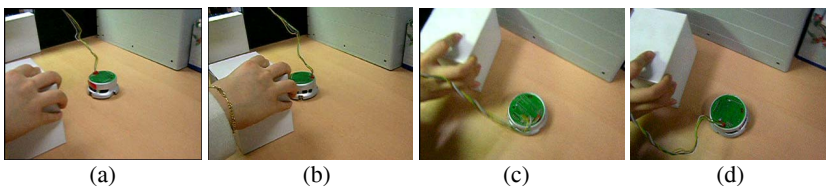


Fig. 6. Avoiding moving obstacle with the real robot. Each shows the avoidance in the middle (a,b) and at the left (c,d) of the robot

References

1. R. C. Arkin, *Behavior-Based Robotics*, MIT Press, 1998.
2. W. D. Smart, *Making Reinforcement Learning Work on Real Robots*, Ph.D.Thesis, 2002.

3. S. Hashimoto, F. Kojima, and N. Kubota, "Perceptual system for a mobile robot under a dynamic environment," *Proc. 2003 IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*, pp. 747-752, 2003.
4. T. Inamura, M. Inaba, and H. Inoue, "User adaptation of human-robot interaction model based on Bayesian network and introspection of interaction experience," *Proc. of the 2000 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2139-2144, 2000.
5. M. N. Nicolescu and M. J. Mataric, "A hierarchical architecture for behavior-based robots," *Autonomous Agents and Multi-Agent Systems*, pp. 227-233, 2002.
6. P. Maes, "How to do the right thing," *Connection Science Journal*, vol. 1, no. 3, pp. 291-323, 1989.
7. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.