Journal of Software Engineering 7 (2): 46-67, 2013
ISSN 1819-4311 / DOI: 10.3923/jse.2013.46.67

SLAM and Path Planning of Mobile Robot Using DSmT

[1]Peng Li, [2]Xinhan Huang, [1]Shengyong Wang and [3]Jean Dezert
[1]WISDRI (wuhan) AUTOMATION Co., Ltd., Wuhan 430205, China
[2]Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[3]ONERA 29 Av. de la Division Leclerc, 92320 Chatillon, France

Corresponding Author: Peng Li, WISDRI (wuhan) AUTOMATION Co., Ltd., Wuhan 430205, China

ABSTRACT

SLAM (Simultaneous Localization and Mapping) and path planning are two important research directions in the field of robotics. How to explore an entirely unknown dynamic environment efficiently is a difficult problem for intelligent mobile robots. In this study, a new method of information fusion i.e. DSmT (Dezert-Smarandache Theory) which is an extension of DST (Dempster-Shafer Theory) is introduced to deal with high conflicting and uncertain information and then multi-agent robot system with GREM (Generalized Evidence Reasoning Machine) based on DSmT is presented for mobile robot's SLAM and efficiently planning smooth paths in unknown dynamic environment. The single robot is treated as a multi-agent system and the corresponding architecture combined with cooperative control is constructed. Considering the characteristics of sonar sensor, the grid map method is adopted and a sonar sensor mathematical model is constructed based on DSmT. Meanwhile a few of gbbaf (general basic belief assignment functions) are constructed for fusion. In order to make the A* algorithm which is the classical method for the global path planning suitable for local path planning, safety guard district search method and an optimizing approach for searched paths are proposed. Finally, SLAM and path planning experiments are carried out with Pioneer 2-DXe mobile robot. The experimental results testify the validity of hybrid DSm (Dezert-Smarandache) model under DSmT framework for fusing imprecise information during map building and also reveal the validity and superiority of the multi-agent system for path planning in unknown dynamic environment.

Key words: SLAM, path planning, multi-agent, dezert-smarandache theory, mobile robot

INTRODUCTION

SLAM (Simultaneous Localization And Mapping) and path planning are two important research directions in the field of robotics. How to explore an entirely unknown dynamic environment efficiently is a difficult problem for intelligent mobile robots. Robot obtains information through its sensors such as sonar, laser, infrared, optical, etc. to construct the map of the environment. Because of the availability and low cost of the sonar, it is the most popular sensors for robot. Maps are used for path planning, point to point navigation and robot localization. The robot SLAM has been extensively explored in last two decades (Yang *et al*., 2005; Minguez and Montano, 2004).

In unknown dynamic environment there are many uncertainties and conflicts that one needs to deal with. The most influencing factor is moving object, because it will cause high conflicting and

uncertain information, for a example, if an object was at a place A and now it has moved to an another place B, then the place A was occupied and now is empty, the place B was empty but now is occupied, this is conflicting information. And the accumulative substantial positional errors occur from the inertial or dead-reckoning navigation, especially for a long distance travel. For robot's SLAM, how to effectively filter these uncertainties is still a problem and a precise environment map cannot be built without solving this difficult problem. A precise map is a strong requirement for a good path planning, since the robot cannot find a rational/optimal way without a precise map of the environment.

Robot SLAM studies can fall into two subgroups as follows: (1) Grid based environment maps, (2) Object specification based maps. Most researches usually use the first method. Noykov improved the method for occupancy grid map (Noykov and Roumenin, 2007). Yenilmez used the Sequential Principal Component (SPC) method to process sonar data, discussed mathematical and experimental issues of the method by comparing a Bayesian method that works efficiently in map building using sensor data (Yenilmez and Temeltas, 2007). Because of the physical characteristics limitation of sonar sensors, the data obtained by sonar often contains unbelievable errors. So recently the laser sensor which is more accurate than sonar sensor becomes more popular. Grisetti *et al.* (2007) presented an approximate but highly efficient approach to mapping with Rao-Blackwellized particle filters. Zhang *et al.* (2008) proposed a robust regression model for segment extraction in static and dynamic environments. But the dynamic factor could be filtered well. Liang *et. al* (2009) make use of cameras for SLAM in structured environment, but effective SLAM in unknown dynamic environment is still a problem. Furthermore, if an excellent method can deal with errors brought by sonar readings, sonar sensors can also be used to paint the dynamic environment maps accurately.

After building an accurate environment map, it is needed to solve the path planning problem in dynamic environment which is composed of two main parts: the global path planning and the local path planning. In most global path planning methods, the robot stores the information of the entire environment, it only needs to calculate the path once at the beginning and then to reach the target point through fixed path. Contrariwise, for the local path planning, the robot knows only the area which has been detected by itself and must choose the direction to move.

Latombe (1991) described most of the methods which solved the path planning problem for mobile robots, such as road map, cell decomposition and artificial potential field methods (APF). Additionally, several recent approaches have been proposed to solve this problem. For example, Zhong *et al.* (2008) proposed a new method based on neural dynamics for robot path planning in a varying environment. But it is only suitable for static environment. Belkhous *et al.* (2005) proposed a new approach for trajectory optimization of a mobile robot in a general dynamic environment. In this study, the experiment scenario is also under static environment. Yin and Yin (2008) proposed an improved potential field method for automatic mobile robot path planning. Though this method can deal with the dynamic environment with moving obstacles, it is only a reaction to the moving obstacles and there is no planned path for mobile robot. There are also some other methods, some of them use global methods to search the possible paths in the workspace (Li and Bui, 1998), usually only dealing with static environments; some are trapped in undesired local minima such as concave U-shaped barriers. There are few valid methods can both deal with dynamic path planning and avoiding moving objects.

To deal with the high conflicting and uncertain information in unknown dynamic environment for robot SLAM and path planning, a brand new information fusion method namely DSmT

(Dezert, 2002; Dezert and Smarandache, 2003) which is an extension of DST (Shafer, 1976) is proposed. DSmT is able to solve complex static or dynamic fusion problems beyond the limits of the DST framework. And then, the kernel of this system GERM (Generalized Evidence Reasoning Machine) with DSmT is proposed. In traditional robot studies, the robot is always treated as a single unit. But here the robot acts as a multi-agent system, such as path planning agent, behavioral agent and perception agent. In this system, agents can be complex entities at the same time. Each agent achieves its task and collaborates with other agents. The client agent is the monitoring center. It constructs the real time environment map with information transferred from the robot and send commands to the robot.

MULTI-AGENT ROBOT SYSTEM ARCHITECTURE

The multi-agent robot system can be divided into five subsystems of agents: perception, behavior, path planning, localization and actuator (Fig. 1). The behavioral agent subsystem includes goto agent and avoid agent. In addition to all of the aforementioned, there is a client agent acting as user interface. Fig. 1 also depicts the flow of information among the different agents.

The perception agent obtains information about the environment and the robot's internal conditions. Of course, it includes an error reading filter based on DSmT. It collects data from the sensors and after getting rid of error readings adapts them to provide the information requested by the other agents of the system. For example, the milemeter is in charge of obtaining the coordinates (x, y) of the robot and its orientation, with reference to a fixed axis; the sonar sensor collects all the sonar readings of the robot.

The localization agent locates the robot on the global map, the path planning agent searches for obstacle-free paths.

The behavioral agent subsystem carries out specific actions, such as avoiding obstacles, going to a point, etc. The information coming from the localization agent and path planning agent is used to react or respond to the changes generated by the robot or the environment. The following agents have been defined: the goto agent which is in charge of taking the robot from the initial to the final coordinates without considering obstacles; the avoid agent which must go around the obstacles when they are found in the path of the robot.

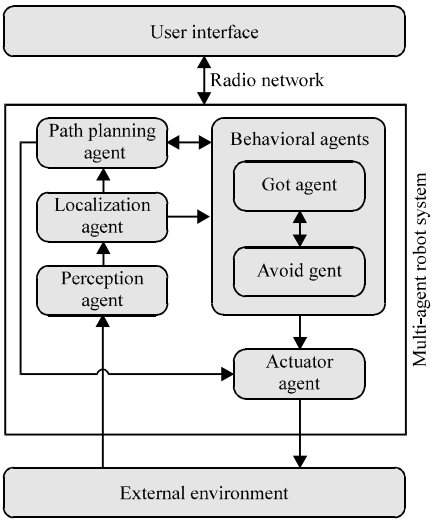

Fig. 1: Multi-agent robot system and the relationships among the different agents

The actuator agent is responsible for directly using the robot's various performance motion components, such as linear and angular velocity controllers, etc.

According to the environmental data transmitted from the robot, the client agent constructs the real time environment model using GERM and gives commands to robot. For example, when all the agents are running, the user can request a task through the client agent which sends the new robot goal to the path planning agent. Then the path planning agent divides the task into a series of turning point goal and sends the first target position to the goto agent. Based on this information and the actual position (obtained from the localization agent), the goto agent calculates the best linear and angular speeds to reach the target. On the other hand, based on the information provided by the localization agent and sonar agent, the avoid agent calculates the linear and angular speeds needed to dodge the obstacle. At this point both agents (goto and avoid) negotiate in order to decide who uses the motors, the negotiation method will be introduced later. But usually the avoid agent does not need to work because the path planning has found a collision free path for robot except accidents, for example, an object abruptly appears in front of the robot and the path planning yet has not calculated the new path for current situation.

The winning one sends the desired speeds to the actuator agent. And then perception agent obtains the sonar and the milemeter readings and correspondingly sends them to the localization agent. With this new information all the agents update their internal state and new decisions can be taken. Once the target position sent by the path planning agent is reached, another target position will be sent to the goto agent.

The robot's status can be monitored by client agent. This agent depicts a real-time global map using DSmT algorithm and controls the robot according to the information transmitted from the robot via radio network.

SIMPLE REVIEW OF DSMT

Although very appealing, the DST presents some weaknesses and limitations (Murphy, 2000) already reported by Zadeh (1979, 1984, 1985, 1986) and Dubois and Prade (1986) in the eighties and reinforced by Voorbraak (1991) in because of the lack of complete theoretical justification of Dempster's rule of combination, but mainly because of low confidence to trust the result of Dempster's rule of combination when the conflict becomes important between sources.

The DSmT of plausible and paradoxical reasoning proposed by Dezert and Smarandache (2004, 2006, 2009) in recent years allows to formally combine any types of independent sources of information represented in term of belief functions. And it is mainly focused on the fusion of uncertain, highly conflicting and imprecise sources of evidence. DSmT is able to solve complex static or dynamic fusion problems, especially when conflicts between sources become large and when the refinement of the frame of the problem under consideration, denoted Θ, becomes inaccessible because of the vague, relative and imprecise nature of elements of Θ.

Notion of hyper-power set D^{Θ}: One of the cornerstones of the DSmT is the notion of hyper-power set. Let $\Theta = \{\Theta_1, \ldots, \Theta_n\}$ be a finite set (called frame) of n elements. The hyper-power set D^{Θ} is defined as the set of all composite propositions built from elements of Θ with $\cup$ and $\cap$ operators such that:

(a) $\Phi, \theta_1, \ldots \theta_n \epsilon D^{\Theta}$
(b) $A, B \epsilon D^{\Theta}$, then $A \cap B \epsilon D^{\Theta}$ and $A \cup B \epsilon D^{\Theta}$
(c) No other elements belong to D^{Θ}, except those obtained by using rules (a) or (b)

The classic Dsm (dezert-smarandache) rule for free-DSm model: When all elements of the frame Θ are considered as partially overlapped, this model is named as free-DSm Model. When some integrity constraints (exclusivity or non-existential constraints among some elements of the frame) are introduced, this model is called as hybrid DSm model.

For k independent uncertain and paradoxical sources of information providing generalized basic belief assignment m(.) over D^{Θ}, the classical DSm conjunctive rule of combination $m_{M^f(\Theta)}(A)$ for working with a free DSm model denoted $M_f(\Theta)$ is given by:

$$\forall A \neq \phi \in D^{\Theta}, m_{M^f(\Theta)}(A) = [m_1 \oplus \cdots m_k](A) = \sum_{\substack{X_1,\ldots,X_k \in D^{\Theta} \\ (X_1 \cap \cdots \cap X_k) = A}} \prod_{i=1}^{k} m_i(X_i) \quad (1)$$

$m_{M^f(\Theta)}(A) = 0$ by definition, unless otherwise specified in special cases when some source assigns a non-zero value to it.

DSm rule of combination for hybrid DSm model: The formula of hybrid DSm rule of combination for working with hybrid DSm models is defined for all $A \in D^{\Theta}$ by Dezert and Smarandache (2004):

$$m_{M(\Theta)}(A) = \phi(A)[S_1(A) + S_2(A) + S_3(A)] \quad (2)$$

$$S_1(A) \equiv m_{M^f(\Theta)}(A) = \sum_{\substack{X_1,X_2,\ldots,X_k \in D^{\Theta} \\ (X_1 \cap X_2 \cap \ldots \cap X_k) = A}} \prod_{i=1}^{k} m_i(X_i) \quad (3)$$

$$S_2(A) = \sum_{\substack{X_1,X_2,\ldots,X_k \in \phi \\ [(u(X_1) \cup u(X_2) \cup \ldots \cup u(X_k)) = A] \vee \\ [(u(X_1) \cup u(X_2) \cup \ldots \cup u(X_k)) \in \phi) \wedge (A = I_t)]}} \prod_{i=1}^{k} m_i(X_i) \quad (4)$$

$$S_3(A) = \sum_{\substack{X_1,X_2,\ldots,X_k \in D^{\Theta} \\ (X_1 \cup X_2 \cup \ldots \cup X_k) = A \\ X_1 \cap X_2 \cap \ldots \cap X_k \in \phi}} \prod_{i=1}^{k} m_i(X_i) \quad (5)$$

where, $I_t = \theta_1 \cup \theta_2 \cup \ldots \cup \theta_n$ denotes the total ignorance.

The conflict among the sets X_i can be directly generalized for $k \geq 2$ sources as follows:

$$k_{12\ldots s} = \sum_{\substack{X_1,\ldots,X_s \in D^{\Theta} \\ X_1 \cap \ldots \cap X_s = \phi}} \prod_{i=1}^{s} m_i(X_i) \quad (6)$$

The DSm hybrid rule of combination (2) increases the non-specificity of the result specially in a sequential/temporal fusion process since the masses of belief coming from partial conflicts are transferred to some partial ignorance. To avoid this main drawback in a sequential fusion process, it is better to redistribute the conflicting masses back to the elements only involved in the conflicts. This is the idea of the new family of Proportional Conflict Redistribution (PCR) rule. Since the redistribution can be done in many ways, there exist several PCR rules. The simplest one satisfying

Table 1: Comparison between PCR2 and PCR5

Rules	Calculation time (msec)	MD of $m(\theta_1)$	MD of $m(\theta_2)$	MD of $m(\theta_1 \cup \theta_2)$
PCR2	5	2.6×10^{-5}	-2.6×10^{-5}	0
PCR5	10			

MD: Mean deviation

the neutrality of the vacuous belief assignment is PCR2 and the most efficient one is PCR5 (Li *et al.*, 2006a). Both are analyzed in application of this study and the PCR2 is chosen in the next paragraph. The PCR2 formula for $k\geq2$ sources is:

$$\forall(X\neq\phi)\in D^{\Theta},\ m_{PCR2}(X)=\sum_{\substack{X_1,X_2,\ldots,X_s\in D^{\Theta}\\ X_1\cap X_2\cap\ldots\cap X_s=X}}\prod_{i=1}^{s}m_i(X_i)+C(X)\frac{c_{12\ldots s}(X)}{e_{12\ldots s}}k_{12\ldots s} \tag{7}$$

$$C(X)=\begin{cases}1, & \text{if X involved in the conflict,}\\ 0, & \text{otherwise;}\end{cases}$$

where, $c_{12\ldots s}(X)$ is the non-zero sum of the column of X in the mass matrix, $k_{12\ldots s}$ is the total conflicting mass and $e_{12\ldots s}$ is the sum of all non-zero column sums of all non-empty sets involved in the conflict.

PCR5 is usually considered as the most efficient fusion rule from the theoretical point of view, however it requires more time than other rules to compute the results and this is the price to pay for a better management of the conflicting information in the fusion process. In this application, the results obtained with PCR5 with respect to those obtained with the simpler PCR2 rule of combination are compared. For such comparison, twenty thousands random groups of basic belief assignments each group/sample was composed of $m_1(\Theta_1)$, $m_1(\Theta_2)$, $m_1(\Theta_1\cup\Theta_2)$, where $m_1(.)$ and $m_2(.)$ corresponds to two sources of evidences have been generated and the two sources of the random samples with PCR2 and PCR5 fusion rules have been combined. Because the PCR5 is more accurate than PCR2 in principle, make the PCR5 fusion results ($m(\Theta_1)$, $m(\Theta_2)$ and $m(\Theta_1\cup\Theta_2)$) to be the standard basic belief assignments, the comparisons between the mean deviations (MD) obtained from PCR2 and PCR5 are shown in Table 1. From Table 1 it is clearly shown that the MDs with PCR2 and with PCR5 are very small and can be ignored. It means that in this case, there is almost no difference between PCR2 and PCR5 in basic belief assignments. The big advantage of PCR2 over PCR5 is that it only took 5ms for PCR2 to calculate 20 thousands groups of random belief assignments, whereas PCR5 requires twice time for carrying out the computations.

Since, in this application, the real-time performance is very important because the robot must perform all calculations (including calculating about 50 thousands groups of belief assignments) in one fusion cycle which is about 100 msec, PCR2 fusion rule instead of the more computational time consuming PCR5 fusion rule has been adopted.

Architecture of ERM (evidence reasoning machine): In Shafer (1976) book, the paradigm shift was stated which led him to formulate an alternative to the existing Bayesian formalism for automated reasoning, thus leading to what is commonly known as Dempster-Shafer (DS) evidential reasoning. After that, many researchers proposed their own theory based on DS theory to deal with the drawbacks of DS theory, among these theories, the Dezert-Smarandache Theory is a recent

appealing alternative for dealing efficiently with high conflicting sources of evidences in practice. But until now, there is no uniform architecture for these theories. So this article proposes an uniform architecture for evidence reasoning theories.

The evidence reasoning theories usually can be divided into four layers: evidence source layer, belief assignment layer, belief fusion layer and decision-making layer. But most of the rules of combination proposed in these theories are based on the assumption of the same reliability of sources of evidence. When the sources are known not being equally reliable and the reliability of each source is perfectly known, it is natural and reasonable to discount each unreliable source proportionally to its corresponding reliability factor. Shafer proposed discounting method to revise the assigned belief. Then a belief revision layer should be added between belief fusion layer and belief assignment layer. The basic framework of ERM is shown in Fig. 2. The belief fusion layer can adopt different algorithms of these evidence theories.

GERM based on DSmT: Because of the advantages of DSmT (Dezert and Smarandache, 2009), it is a good choice for ERM to make use of this theory. The GERM with two elements (θ_1, θ_2) based on DSmT is shown in Fig. 3. In the belief assignment layer, the F(S) is the belief assignment function, through which the original Basic Belief Assignments (BBA) O(.) are calculated. In the belief revision layer, instead of discounting method proposed by Shafer, the ANN (artificial neural

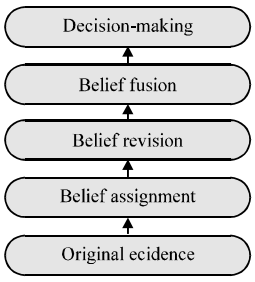

Fig. 2: Basic framework of evidence reasoning machine

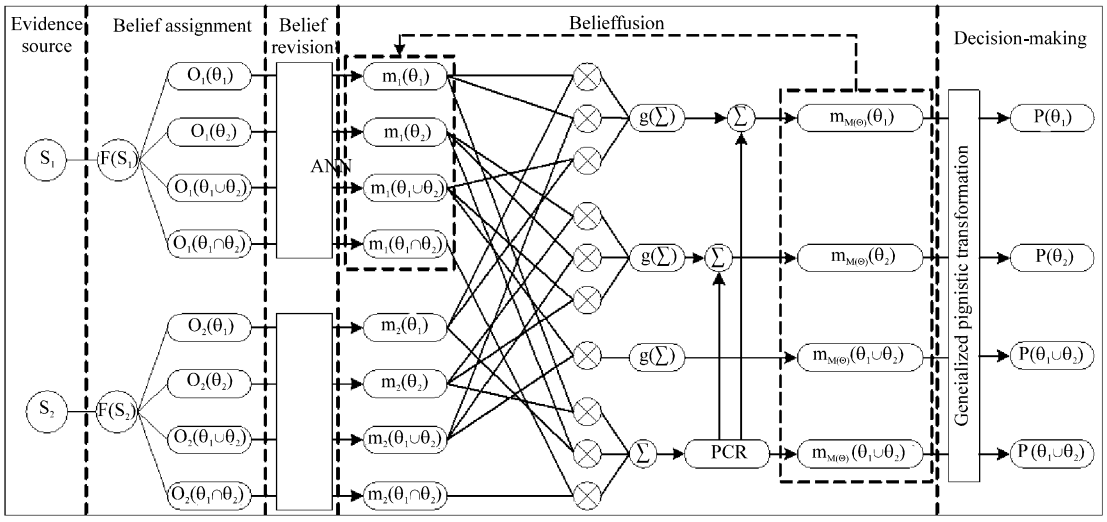

Fig. 3: Structure of GERM (generalized evidence reasoning machine) based on DSmT

network) is adopted. For under most conditions, the discounting values are unknown and nonlinear, but Shafer's discounting method is linear and the discounting values are fixed. The ANN is more flexible, it can learn of the nonlinear discounting curves through the foregone data, so it can depict the discounting situations more precisely than Shafer's method. The m(.) is the bba revised by ANN. ⊗ is multiplication operator. The g (Σ) is actuating function. For DSmT, this function is a simple sum operator; it can be rewritten for other evidence theory such as DST. $m_{M(\Theta)}(.)$ is the result of fusion. The PCR is proportional conflict redistribution rule. The conflict factor can be reassigned to $m_{M(\Theta)}(\theta_1)$ and $m_{M(\Theta)}(\theta_2)$, but if the conflict factor is not needed to be reassigned, it will directly become the $m_{M(\Theta)}(\theta_1 \cap \theta_2)$. The decision-making layer is an expanded layer if need. The probabilities of elements can be calculated through Generalized Pignistic Transformation (GPT). Usually this layer is not necessary. For the whole machine, though there are only two input evidence source, it can deal with multi-source instances. Suppose a finite set of n sources $\{X_1,\ldots,X_n\}$, in the beginning X_1 and X_2 is treated as S_1 and S_2. After the first belief fusing cycle, the $m_{M(\Theta)}(.)$ is got, but it is not the final result, so the $m_{M(\Theta)}(.)$ returns and becomes the m(.) (it is treated as another S_1 and directly enters the belief fusion layer) and then X_3 participate in fusion, it is treated as another S_2. After belief assigning and revising, the m(.) computed from X_3 will be fused with the m(.) converted directly from last fusion result $m_{M(\Theta)}(.)$ and their fusion result will be treated as a source in the next fusion cycle. All of the sources can be processed in sequence. When the final fusion result comes out, it can enter the final decision-making layer.

Under the two elements θ_1 and θ_2 condition, this GERM includes the classical DSm model and the hybrid DSm model. When there is no integrity constraint in the model of the frame, the hybrid rule of combination 2 reduces to classical rule because $S_2(A)$ and $S_3(A)$ in Eq. 2 become zero. Off course if the amount of elements is more than 2, the architecture of the belief fusion layer in this GERM is more complex. But under most conditions two elements are enough.

This GERM is also suitable for DST. In DST situation condition, the $O(\theta_1 \cap \theta_2)$ and $m(\theta_1 \cap \theta_2)$ are not needed, their values are zero. The actuating function $g(\Sigma)$ should be changed into DST fusion rule. For other evidence theories, this structure of GERM only needs to change the architecture of belief fusion layer a little.

Mathematic sonar sensor model based on DSmt: The simple principle of sonar sensor is: it generates sheaves of cone-shaped wave to detect the objects. The wave will be reflected as soon as it encounters an object. Since the sound wave generated by sonar sensor spreads forwards in the form of loudspeaker, there exists a divergence angle. Any object in the fan-shaped area can reflect the wave, so the real position of the objects detected among the fan-shaped area cannot be clearly known. And there are many other environment influences, so the data obtained by sonar sensor is not accurate. The range of sonar readings is from 10 cm to nearly 5 m.

According to the characteristics of sonar sensor, a mathematic model of sonar sensor (Fig. 4.) based on DSmT has been constructed. Suppose there are two elements θ_1 and θ_2 in the frame of discernment Θ. Here the Θ is defined as the status of each grid on the map constructed by the robot.

θ_1 means the grid in map is empty, θ_2 means occupied by some objects, $\theta_1 \cap \theta_2$ means the grid cannot be clearly ascribed to empty state or occupied state, it is the conflict mass generated by the computation of information fusion and $\theta_1 \cap \theta_2$ means that information about the grid is unknown. The hyper-power set of the discernment frame Θ is $D^{\Theta} = \{\phi, \theta_1 \cap \theta_2, \theta_1, \theta_2, \theta_1 \cap \theta_2\}$. Then $m(\theta_1)$ denotes the General Basic Belief Assignment Functions (GBBAF) for the empty status, define $m(\theta_2)$ as the

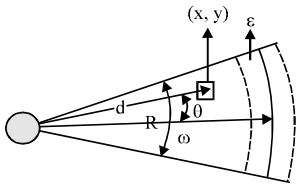

Fig. 4: Sonar model based on DSmT

gbbaf for occupied status, $m(\theta_1 \cap \theta_2)$ is considered as the gbbaf of conflict mass and $m(\theta_1 \cap \theta_2)$ is defined as the gbbaf of unknown status (it mainly refers to those areas that are still not scanned at present).

Though Li has achieved some success in this field (Li *et al.*, 2006b, 2007), but the constructed formulas cannot deal with dynamic environment and cannot distribute the conflicting mass accurately. The new belief assignments $m(.): D^{\Theta} \rightarrow [0, 1]$ are constructed by authors such as the Eq. 8-12 according to the sonar based on DSmT:

$$m(\theta_1) = \left(1 - \frac{\lambda}{2}\right) \cdot \exp\left[\frac{-d^2}{2(R\rho_E)^2}\right], \quad R_{min} \le d \le R + 2\varepsilon \tag{8}$$

$$m(\theta_2) = \lambda \cdot \exp\left[\frac{-(d-R)^2}{2R\rho_O^2}\right], \quad R_{min} \le d \le R + 2\varepsilon \tag{9}$$

$$m(\theta_1 \cap \theta_2) = \exp[-\rho_C(\ln d - \beta)^2], R_{min} \le d \le R + 2\varepsilon \tag{10}$$

$$m(\theta_1 \cup \theta_2) = \begin{cases} (1-\lambda) \cdot \tanh\{\frac{\rho_I[d-(R+\varepsilon)]}{R}\}, R + \varepsilon \le d \le R + 2\varepsilon \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{, other} \quad ; \end{cases} \tag{11}$$

$$\beta = \ln\left(\frac{\rho_E\sqrt{R^3}}{\rho_E\sqrt{R} + \rho_O}\right) \tag{12}$$

$$\lambda = \begin{cases} 1 - (\frac{2\theta}{\omega})^2 & 0 \le \theta \le \frac{\omega}{2} \\ 0 & \text{other.} \end{cases} \tag{13}$$

Equation 13 is given by Elfes (Moravec and Elfes, 1985). Here R_{min} is the minimum sonar reading that sonar can obtain. R is the sonar reading between sonar and object. d is the distance from sonar to the point (x, y) in the map. ω is the divergence angle defined as 15 degrees. θ is the angle between d and axis. ε is the error of sonar reading. ρ_E, ρ_O, ρ_C and ρ_I are the environmental adjustment coefficients. Here, $\rho_E = 0.35$, $\rho_O = 0.1$, $\rho_C = 80$ and $\rho_I = 10$.

Then experiments of sonar reading errors are carried out to analyze the ε of the mathematical sonar model. The results are shown in Table 2. An object is placed in front of a fixed sonar sensor to survey the sonar readings. 20 sonar readings are recorded every 500 mm. The distance between object and sonar sensor is from 500-3000 mm. The means of errors between actual distance and

Table 2: Experimental sonar readings at different distances and relevant analysis

Actual distance R_a(/mm)	Mean of errors e_m (mm)	$\lvert e_m/R_a \rvert \times 100$ (%)
500	1.70	0.34
1000	-2.50	0.25
1500	-5.35	0.36
2000	-7.10	0.36
2500	-6.40	0.26
3000	-11.70	0.39
3500	-21.30	0.61
4000	-25.20	0.63

sonar readings are calculated in Table 2 as well as error rates. It is obviously that the error rates in the range of 3000 mm are much smaller than the error rates out the range of 3000 mm. It is proved that in the range of 3000 mm the sonar readings are much more accurate than those out the range of 3000 mm. All error rates in the range of 3000 mm are under 1%. So define the valid maximum R is 3000 mm and ε equals to 1%×R.

Considering the working principle of sonar, the smaller the location of sonar reading is, the greater the possibility of being occupied. Therefore the gbba (general basic belief assignment) of m(θ1) near the sonar reading R is much greater than any other places. In other words, the grid near R is probably occupied by some objects. But the gbba of $m(\theta_2)$ is opposite. In the area near R the gbba of $m(\theta_2)$ almost becomes zero and in the sonar range, the farther away from R the greater the possibility of being empty. The conflicting mass $m(\theta_1 \cap \theta_2)$ is an uncertain part generated in the application of using hybrid DSm rules, it cannot be clearly ascribed to $m(\theta_1)$ or $m(\theta_2)$ and becomes greatest in the intersection of $m(\theta_1)$ and $m(\theta_2)$. At last, the gbba of $m(\theta_1 \cup \theta_2)$ becomes large in the areas beyond sonar range. It proves the group of formulas which accords with the characteristic of sonar sensor-the farther the object is, the more uncertainty. The 3D gbba distributing is shown in Fig. 5.

False reflections: Sometimes, one sensor reading is not reliable to be used in localization since it may be a false reflection. To overcome the false reflections, the distance information obtained from adjacent sonar sensors is combined. Any set of adjacent sensors give readings within a predefined tolerance are considered to be reflected from the same object.

When the sonar beam is reflected away from an object, bounced off another object in the robot's environment and detected back by the sonar receiver, then the reading obtained by the sensor which represents the round trip distance of the sonar beam, does not represent the distance between the sensor and the first detected object. In other word, the first detected object appears to be at a distance farther than the actual one. The false reflection is shown in Fig. 6, in order to eliminate the false reading d_3 by comparing it with the adjacent reading d_2. This is achieved as follows:

- If $\lvert d_3-d_2 \rvert < \Delta$, where Δ is a threshold value $\Delta = \lvert d_1-d_2 \rvert$, then d_3 is not a false reflection and in this case $d_3 = a_1$
- If $\lvert d_3-d_2 \rvert > \Delta$, then d_3 is a false reflection and it will be filtered

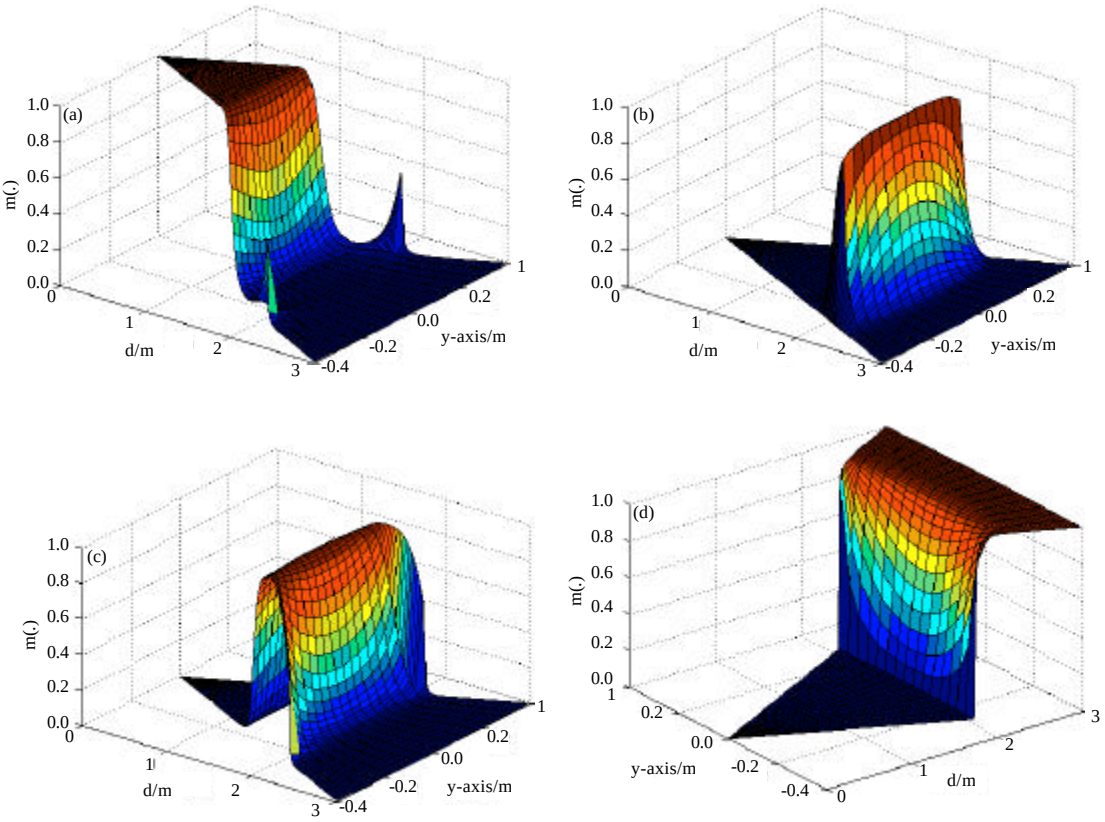

Fig. 5(a-d): 3D gbba distributing when R = 2 m (a) gbba of $m(\theta_1)$ when R = 2 m, (b) gbba of $m(\theta_2)$ when R = 2 m, (c) gbba of $m(\theta_1 \cap \theta_2)$ when R = 2m and (d) gbba of $m(\theta_1 \cup \theta_2)$ when R = 2 m

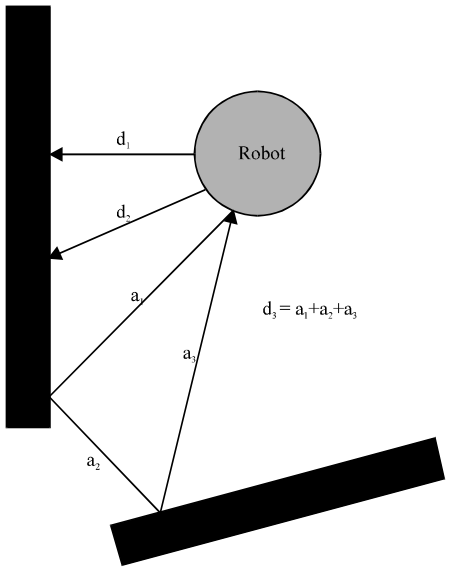

Fig. 6: False reflection of sonar readings

METHOD FOR SELF-LOCALIZATION

Self-localization is very important for mobile robot. If the localization is not precise enough, the error even distortion will appear in the coordinate of the map built by robot. For many effects of environmental factor, precise localization is very difficult. So far, there are many algorithms are applied to self-localization.

Here a simple very efficient method for robot's self-localization is adopted. It is known that the slipping between robot's driver wheels and the ground brings the localization errors. After many experiments, it is found that the main error is the angle error and the value is limited in the range of ±5°. So it is only needed to revise the angle error.

The method of this study is shown in Fig. 7. When the robot gets a group of new data at time t+1, it calculates the belief assignment $m(\theta_2)$ of each grid in the range that sonar can detect and then draws current environment map. The current map only includes the objects detected at time t+1 and the whole map (it includes all objects) which is built at time t is considered as the old map. Then the robot's center is treated as the origin point and the current map is rotated clockwise or anticlockwise with an angle η ($-5 \le \eta \le 5$) around it. During rotations, the match point of the two maps is defined as a point whose $m(\theta_2)>0.8$ in the current map and at the same time the $m(\theta_2)>0.8$ at the same place in the old map. The angle η with greatest amount of match points is treated as the current angle error. And then the robot can revise its internal coordinate according to the η.

Path planning agent: The robot plans the path through path planning agent. After updating the environmental information, the robot firstly makes use of evolutional A* algorithm (Chiang *et al*., 2007) to re-calculate the path if needed. The path from current location to the target point is divided into a series of turning goal points. The A* algorithm is usually used for static global path planning. There are few applications for dynamic local path planning in real time because of its large amount of calculations. In this study, a simple but very efficient method is proposed to make the A* suitable for dynamic local path planning in real time. In most path planning studies, the robot is abstracted as a point without acreage. But actually the robot's radius must be considered in practical applications. The area near the planned path is the safety guard district. That means this area must be empty or the robot cannot pass. The safety guard district is shown in Fig. 8.

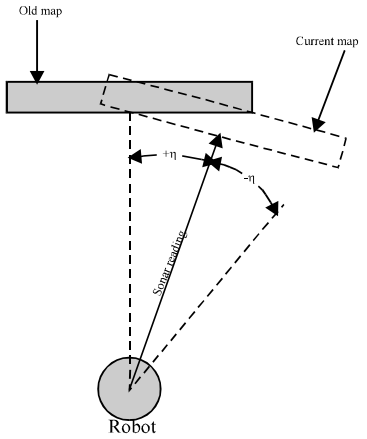

Fig. 7: Robot's self-localization with angular correction

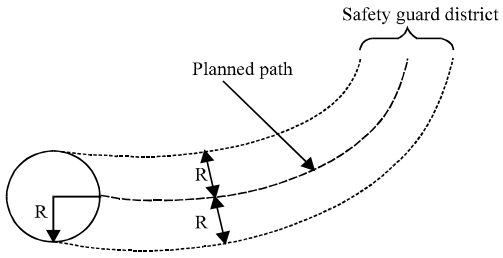

Fig. 8: Safety guard district for robot

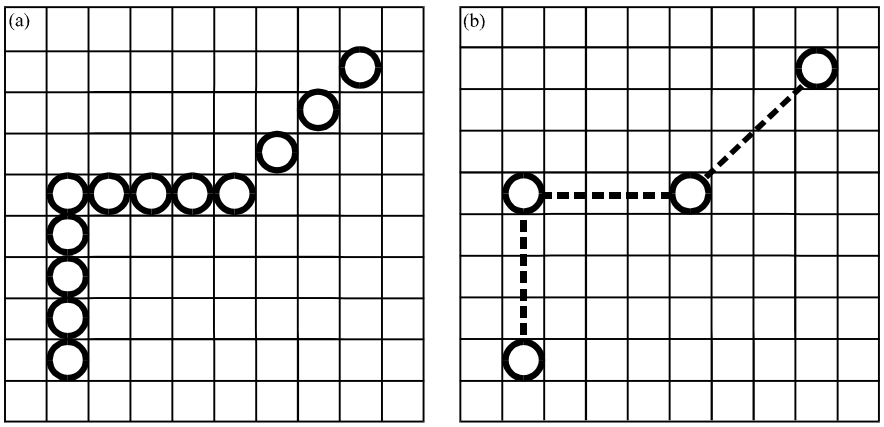

Fig. 9: Sketch map of optimizing point-path (a) Before optimizing and (b) After optimizing

Once the robot gets a new group of sonar readings, each reading is checked to make sure whether it is in the safety guard district. If a reading is in this area, it means an obstacle blocks the way and the path must be re-calculated. Otherwise, if there is none in the area, the path does not need to be re-calculated. This method reduces the computations for the path planning. The robot only needs to calculate the path occasionally.

The path searched by A* algorithm is a group of continuous goal points. If the grid is small, it will spend a large of memory to store the point-path and if the goal points are placed too closely, the robot yet cannot follow the path well because of the limit of its turning radius. So here the point-path is optimized. The goal points on the same line are deleted and then the robot only needs to store the turning goal points (include start point and end point). This method reduces drastically the memory for storing point-path. The sketch map of optimizing point-path is shown in Fig. 9.

Behavioral agents

Gto agent: It is known that the turning angle of the path calculated by evolutional A* algorithm in the grid map is 45 or 90°. So the path is not smooth and sometimes the robot cannot follow the trajectory because of these stark turnings. The goto agent can solve this problem. The robot's walking between two neighbor goal points is under the charge of goto agent.

Avoid agent: When a moving object suddenly appears in front of the moving robot, the avoid agent will take charge of the robot to avoid colliding with the object. Usually this agent does not need to work because the path planning agent can often recalculate the new path without collision

quickly. So the avoid agent only works when the path planning agent has not recalculated the path. The classical algorithm (namely APF) is adopted in the avoid agent. In the traditional APF, an obstacle is considered as a point of highest potential and a goal as a point of lowest potential. In the domain of robot path planning, a robot always moves from a high potential point to a low potential point.

EXPERIMENTS

A user interface as a software platform for experiment is developed with Visual Studio 2008. The Pioneer II mobile robot which is used in experiments has 16 sonar sensors fixed around it. Each sonar sensor is a source of evidence. First of all, the discounting function for each sonar should be learnt through ANN. Take a sonar data 502 mm for example, the sonar reading is treated as the midline of the sonar range ($\lambda = 0$). The object is surveyed at the distance of 502 mm but the actual distance is 500 mm. Then four BBA for the distance of 500 mm are calculated Eq. 7-10, R = 0.502 and d = 0.5, the BBA are $m(\theta_1) = 0.0086$, $m(\theta_2) = 0.9913$, $m(\theta_1 \cap \theta_2) = 0.0001$ and $m(\theta_1 \cup \theta_2) = 0$) and the actual bba for the distance of 500mm are $m(\theta_1) = 0$, $m(\theta_2) = 1$, $m(\theta_1 \cap \theta_2) = 0$ and $m(\theta_1 \cup \theta_2) = 0$. Then the vector [0.0086, 0.9913, 0.0001, 0] is the input of the ANN and the target output is [0, 1, 0, 0]. The data of sonar readings can train an ANN for a sonar sensor. 16 ANN should be trained here for there are 16 sonar sensors.

There are many excellent ANN algorithms, but here a complex one is not needed and then the classical BP neural network with three layers is adopted. Both input layer and output layer have four neurons. The hidden layer includes five neural neurons. The training function is Levenberg-Marquardt back propagation because it is very fast. The activation function is the sigmoid function. The performance function is Sum Squared Error (SSE). The learning rate is 0.1 and the error goal is 0.001. In order to enhance the precision, the input and output vectors are magnified 1000 times.

After training of BP neural network, another group of sonar readings are used to testify the validity of revision. The comparison between gbba without revision and gbba revised by BP neural network is shown in Table 3, the error rates and the average error rates are calculated separately. It is obviously shown that the error rates with revision are much smaller than the error rates without revision as well as the average error rates. The gbba revised by BP neural network have been normalized. m(.) is the gbba of the grid at the actual distance. It is obviously that the BP neural network performs well.

Table 3: Comparison between gbba without revision and gbba with revision

	R = 502 d = 500	R = 996 d = 1000	R = 1493 d = 1500	R = 1988 d = 2000	R = 2495 d = 2500	R = 2987 d = 3000	AER
ER of $m(\theta_1)$							
No revision	0.0086	0.0081	0.0079	0.0076	0.0075	0.0071	0.78%
Revised	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.01%
ER of $m(\theta_2)$							
No revision	0.9912	0.9867	0.9680	0.9407	0.8991	0.8722	5.70%
Revised	1.0000	1.0000	0.9999	0.9999	1.0000	0.9999	0.01%
ER of $m(\theta_1 \cup \theta_2)$							
No revision	0	0	0	0	0	0	0
Revised	0	0	0	0	0	0	0
ER of $m(\theta_1 \cap \theta_2)$							
No revision	0.0001	0.0053	0.0242	0.0517	0.0934	0.1207	4.92%
Revised	0	0	0	0	0	0	0

ER: Error rate, AER: Average error rate, R: Sonar reading (mm), d: Actual distance (mm)

SLAM experiments: Two experiments have been carried out, respectively for GERM with DSmT and GERM with DST which is the classical evidence reasoning algorithm. First of all, an experiment field (size: 4840×3100 mm) is constructed.

Experiment based on DSmT: In this experiment, the robot must construct the unknown dynamic environment model. Before exploring the environment, the map is assumed entirely unknown and thus one takes $m(\theta_1 \cup \theta_2) = 1$ for each grid cell and $m(\theta_1) = m(\theta_2) = m(\theta_1 \cap \theta_2)$ 0. The fusion steps in one fusion cycle are as follows:

- Robot begins to explore the dynamic environment and send data to monitoring center. Let's suppose that at time t, the, monitoring center has calculated the gbba $m_{M^t(\oplus)}(\theta_1)$, $m_{M^t(\oplus)}(\theta_2)$ and $m_{M^t(\oplus)}(\theta_1 \cup \theta_2)$ for each grid. The conflicting mass has been distributed to $m_{M(\oplus)}(\theta_1)$ and $m_{M(\oplus)}(\theta_2)$, so that the conflicting mass equals to zero
- Then at time t+1 (the next time step), through Eq. 8-13, monitoring center calculates the gbba $m(\theta_1)$, $m(\theta_1 \cap \theta_2)$ and $m(\theta_1 \cup \theta_2)$, for the grid which is being detected at that moment
- Apply the arithmetic of restrict spreading to fuse the information in the fan-shaped areas through combination rules of hybrid DSm model and obtain new gbba values $(m_{M^{t+1}(\oplus)}(\theta_1)$, $m_{M^{t+1}(\oplus)}(\theta_2)$, $m_{M^{t+1}(\oplus)}(\theta_1 \cap \theta_2)$ and $m_{M^{t+1}(\oplus)}(\theta_1 \cup \theta_2)$ for detected grids
- Redistribute the conflict mass $m_{M^{t+1}(\oplus)}(\theta_1 \cap \theta_2)$ to $m_{M^{t+1}(\oplus)}(\theta_1)$ and $m_{M^{t+1}(\oplus)}(\theta_2)$ with PCR2
- Update map data

In the experiment, a person walks in the field all the time to represent the dynamic factor. The demonstration of dynamic map building is shown in Fig. 10. Pictures on the left side are photos of

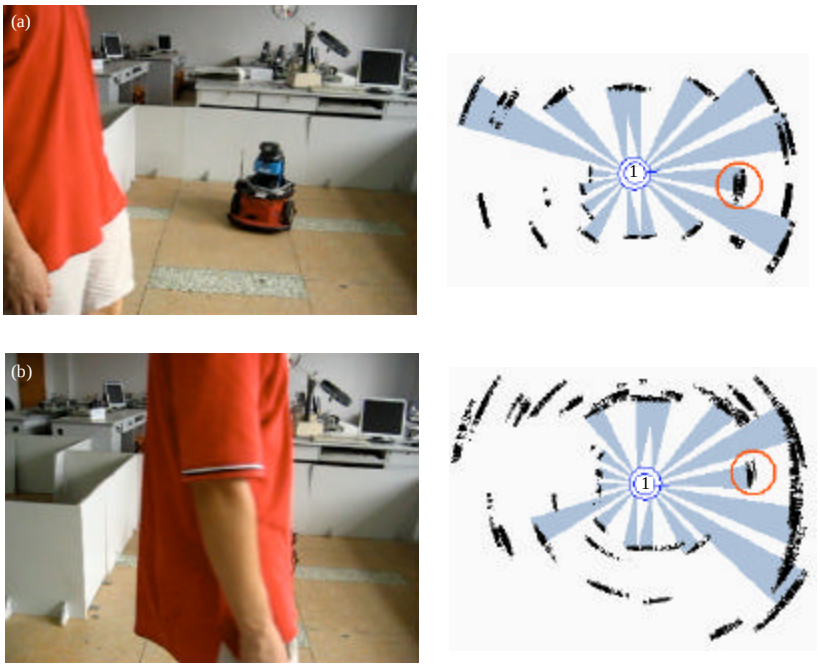

Fig.10(a-b): Moving object detection with DSmT (the moving person is detected by sonar sensors)

real world and figures on other side are the corresponding maps built by monitoring center. In Fig. 10, it is shown that the moving person is detected by sonar sensors of the robot. The locations of the moving person are marked by circles. Obviously, the map is dynamically updated in time. That is to say, if the person stands at location A, the grid of this location is marked as occupied status on the map, but when the person moves to another location B, instead of occupied status the grid of A will marked as empty status, at the same time grid of B will be marked as occupied status.

Experiment based on DST: In this experiment, the mathematical sonar model is the same as in the DSmT-based experiment. But the belief assignment functions only use $m(\theta_1)$, $m(\theta_2)$ and $m(\theta_1 \cup \theta_2)$a because in DST θ_1 and θ^2 are independent elements without conflicting. The degree of conflict is computed during the fusion process. The fusion steps are the same as in the DSmT experiment but in the reassignment of the conflicting mass $m(\theta_1 \cap \theta_2)$.

The experiment based on DST in dynamic environment is shown in Fig. 11. The areas A and B are static objects in real world. The area C is a moving person. Though the sonar has detected the objects in these three areas, after fusion based on DST, the robot estimates that there is nothing in these three areas.

Comparisons of results: The final experimental results are shown in Fig. 12. Left figures are the results of DST and right figures are the results of PCR2 under DSmT. In these figures, the grayscale of the grid point reflects the gbba. The grayscale is calculated as (1-gbba)×255. For the bba maps of $m(\theta_1 \cup \theta_2)$ in the two experiments are nearly the same, here they are not displayed. The final result map is calculated in this way: if the $m(\theta_1 \cup \theta_2)$ of a grid cell is less than 0.5 and m $m(\theta_2) > m(\theta_1)$, this grid cell will be marked as occupied in the final result map.

Path planning experiments: Two kinds of experiments have been carried out: one uses the multi-agent robot system with GERM, the other uses APF which is the classical algorithm for local path planning (i.e., the algorithm of path planning agent is APF). SLAM for the two experiments adopts GERM with DSmT. The initial positions of robot and moving object in the real world are shown as Fig.13. Actually, the moving object is another robot which goes across the field (it acts as a disturbing factor). The barrier in the front of the robot is a U-shape board and the target goal is on the right side of the board.

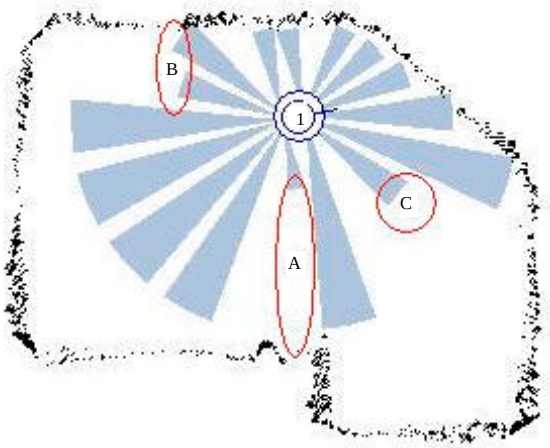

Fig. 11: Experiment result with DST after the combination of sonar readings

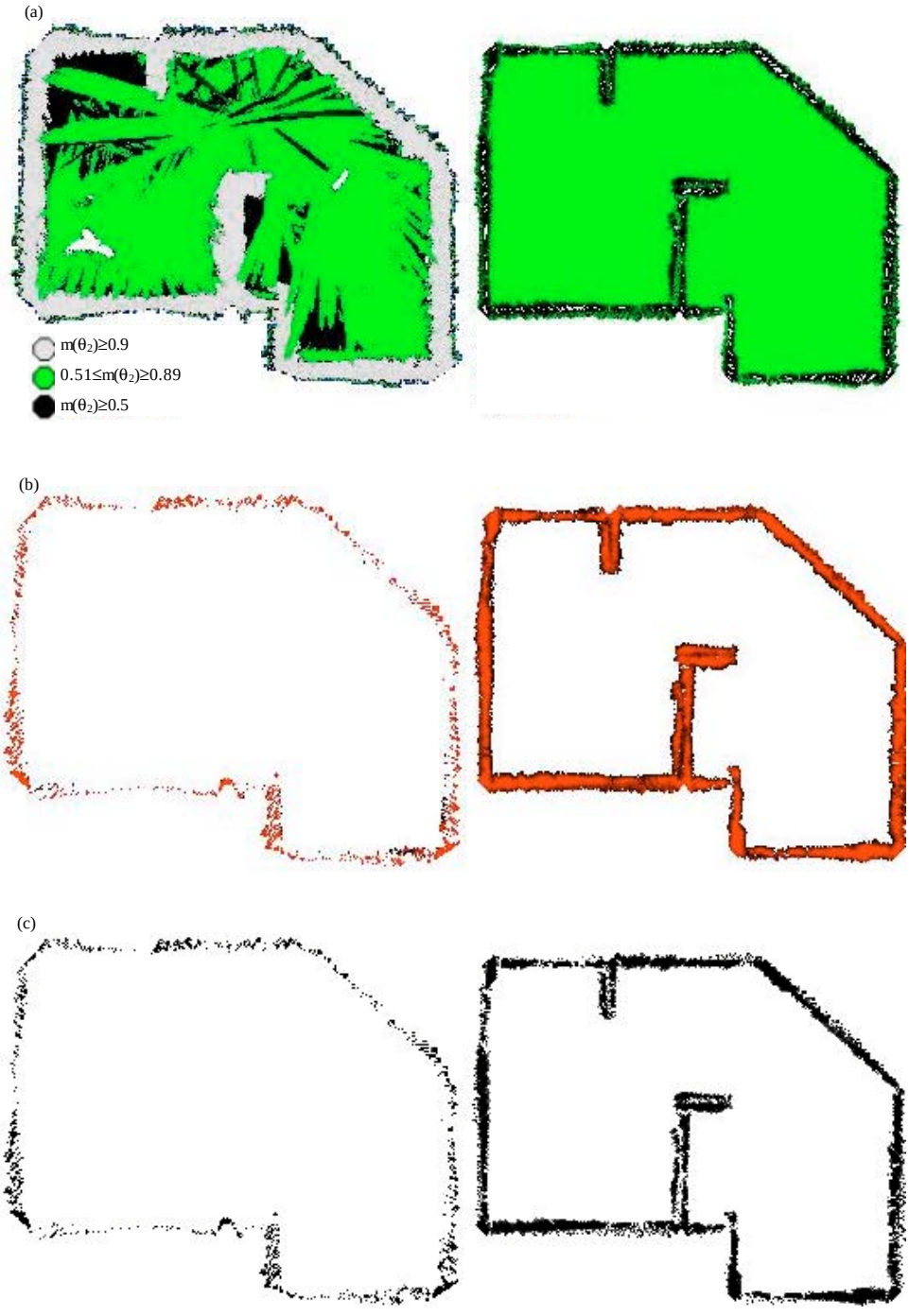

Fig. 12(a-c): Comparison between DST (left subfigures) and DSmT (right subfigures) (a) gbba map of $m(\theta_1)$ (empty status) after the combination of sonar readings, (b) gbba map of $m(\theta_2)$ (occupied status) after the combination of sonar readings and (c) Final result map after information fusion

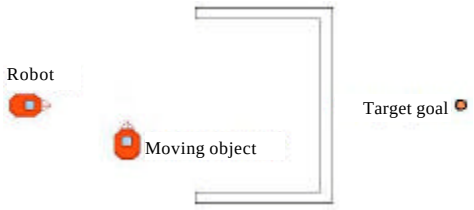

Fig. 13: Initial positions of robot and moving object

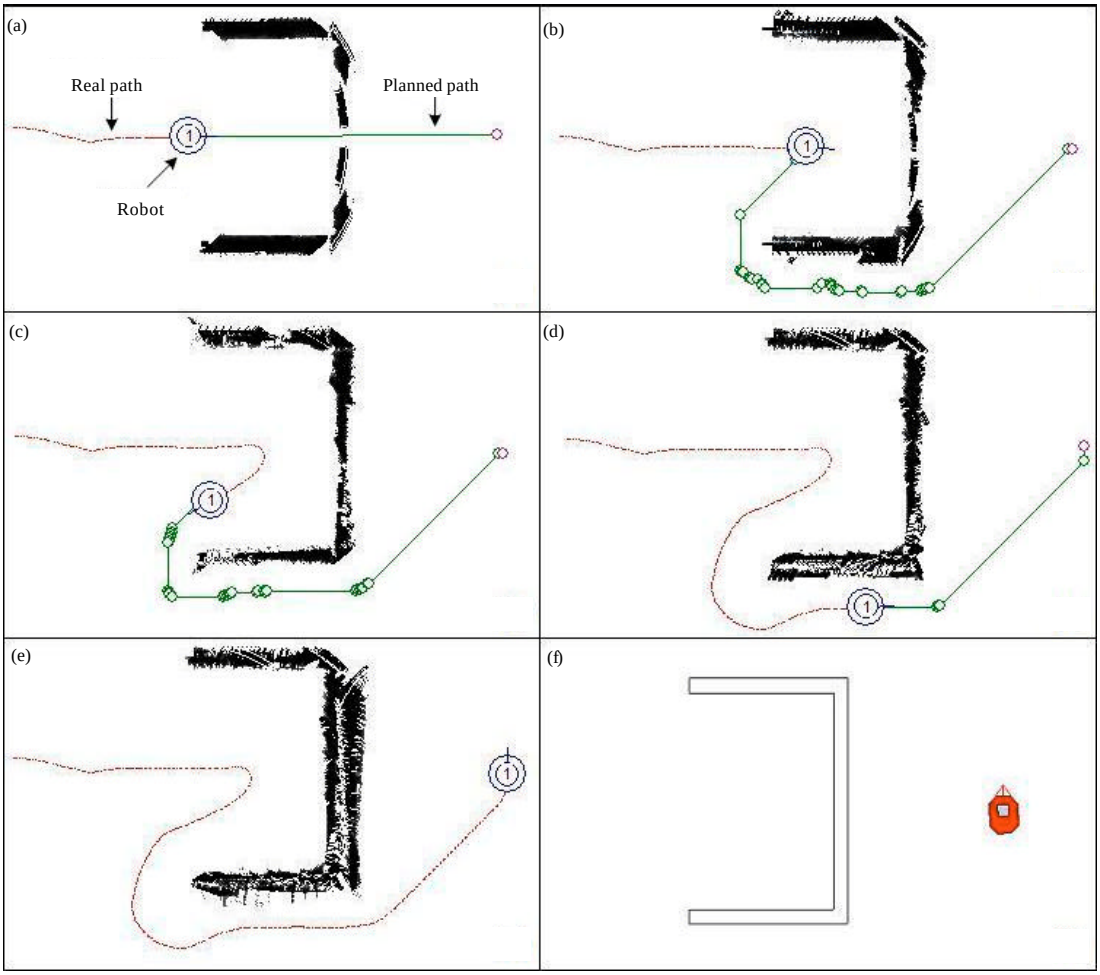

Fig.14(a-f): The robot using GERM can get out from U-shape trap easily, the figure (f) Final status of the robot

Experiment of multi-agent robot system with GERM based on DsmT: The U-shape barrier is a classical barrier for path planning, because many path planning algorithms usually cannot get out from it. The small circles in these figures are the turning goal points calculated by evolutional A* algorithm, the real path and the planning path are also displayed in these figures. The experiment result of multi-agent robot system with GERM based on DSmT is shown in Fig. 14.

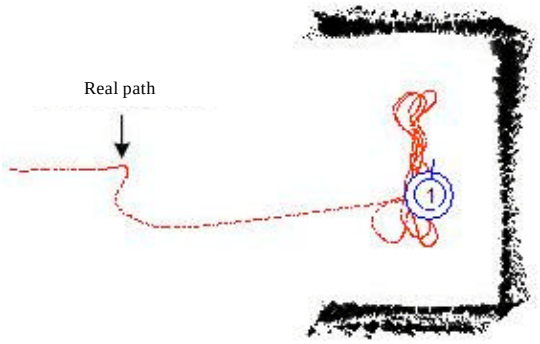

Fig. 15: The robot using APF is trapped in the U-shaped barrier

APF experiment: In this experiment the robot is not a multi-agent system any more. This experiment does not need the behavioral agent subsystem. And then the information flow becomes unilateral. There are no reciprocities between the parts in the robot. So the robot is treated as a single part as most studies. The APF experimental result is shown in Fig. 15. It is shown that the robot is trapped in the U-shaped barrier.

Analysis: Through these two experiments, it is obvious that the multi-agent robot system with GERM based on DSmT is a very efficient method for SLAM and path planning in dynamic environment.

DST has an inherent limitation; it cannot deal with high conflicting and uncertain information and usually leads to a counterintuitive result. The high conflicting and uncertain information come from two aspects: (a) the intersection area of $m(\theta_1)$ and $m(\theta_2)$ shown in Fig. 4, where is the sonar error range ε, is the high conflicting information area, so DST usually concludes that there is nothing in these areas, the objects which is not big enough will be covered with these areas and be marked with empty status. The area A and area B which are described as empty areas in Fig. 11 belong to this situation and (b) in dynamic environment experiment, the moving person in the field is a source of high conflicting and uncertain information. When one stays at a point, the robot marks this grid point with occupied status ($m(\theta_2)$ is much greater than $m(\theta_1)$), once the person leaves this point, the information obtained by the robot shows that this grid point is empty ($m(\theta_1)$ is much greater than $m(\theta_2)$) and then the empty status conflicts with the foregoing occupied status. In Fig.11, the moving person is detected by a sonar sensor (area C), but after being fused with original map information, the area C is considered as an empty region. It is obvious that this result is counterintuitive. But the GERM with DSmT can efficiently deal with these high conflicting and uncertain information. In unknown dynamic environment, the hybrid DSm model under DSmT framework can draw accurate map after filtering the dynamic interferential factors and update the map in time (in Fig.10, it is obvious that the map is dynamically updated in real time). Outlines of objects and location are expressed more clearly. So it is proved that in the DSmT is better than DST (Shafer, 1976) in the area of map building in dynamic environment. And the algorithm in this study can deal with moving object better than previous studies (Zhang *et al.*, 2008; Liang *et al.*, 2009).

Figure 12a reflects the substantial difference between DST and DSmT approaches. $m(\theta_1)$ of DST experimental result marks the actual outlines of walls with high empty gbba values. But the

$m(\theta_1)$ values of those real empty areas in the field are not the same high as the walls'. Furthermore, some real empty areas only have very low $m(\theta_1)$ values. Whereas, $m(\theta_1)$ obtained in DSmT-based experiment depicts the real situations well.

In Fig.12b, $m(\theta_2)$ gbba map of DST experiment is the outer edge map of the whole field, whereas $m(\theta_2)$ gbba map of DSmT experiment is more accurate.

The maps in Fig.12c are the final fusion results of $m(\theta_1)$, $m(\theta_2)$ and $m(\theta_1 \cup \theta_2)$. Clearly, the environment map built by GERM with DSmT is much more accurate than the map built by GERM based on DST. Even some information is falsely described (Fig. 11a, b). And the map built by GERM with DSmT using sonar sensor is almost as accurate as the map built through laser sensor (Grisetti *et al*., 2007).

The localization method is very effective and remains very simple to implement. There is few coordinate errors in the resulting maps. So the map can be built correctly and the map is more accurate than the map built by Yenilmez and Temeltas (2007).

In experiment of multi-agent robot system with GERM based on DSmT, the robot only needs to store several turning goal points. The planned path is changed once the robot finds that new barriers block the way; the new multi-agent method proposed in this study is better than APF method for processing dynamic environment with moving objects. In the whole phase of avoiding moving object, the robot can always find the rational path to the target in real time, so the robot avoids the moving object easily without any excrescent route, it is better than using pure APF method to plan path (Yin and Yin, 2008). But in APF experiment, the avoiding is not so successful. In Fig.15, it is obviously shown that the robot makes a big turn to avoid the moving object. It proves that the method in this study is more accommodative than the method proposed by Zhong *et al*. (2008) in dynamic environment with moving objects.

The goto agent performs so perfectly that the real path of hybrid multi-agent robot system experiment is very smooth instead of stark turns which is the fault of A* algorithm.

Actually, the start position of the robot is a concave U-shaped trap. In multi-agent robot system experiment, the robot can easily walk out this area and the path is rational. But in APF experiment, the robot is trapped in a local minima; it repeatedly follows the same trajectory and cannot get out from the repetition. It proves that multi-agent robot system with GERM based on DSmT can plan better path than APF (Latombe, 1991) or other methods (Li and Bui, 1998).

In multi-agent robot system experiment, the Safety guard district search method reduces the computation, so that the whole system works well without lags and even without any crashes. This experiment based on DSmT was very successful and promising for further developments.

In summary, the multi-agent robot system with GERM based on DSmT can deal with complex dynamic environment with moving objects. It can get rid of the interference of moving object and precisely build the map with the imprecise data obtained by sonar sensors. And this system also has good performance on path planning. It can easily avoid moving object and reach the target through a rational path without trapping in undesired local minima. the SLAM and path planning method proposed in this study is more efficient than previous studies.

CONCLUSION

This study has proposed a novel multi-agent robot system with GERM based on DSmT for robot SLAM and planning smooth paths in an unknown dynamic environment. The single robot is divided into several synergic agents. With the application of hybrid DSm model under DSmT framework, the GERM is constructed, also a group of gbbaf functions is proposed for sonar sensor.

In order to make A* algorithm suitable for dynamic local path planning, the safety guard district search method with an optimizing approach for searching the path are proposed. Through the results of experiments carried out with Pioneer 2-DXe mobile robot, the multi-agent robot system with GERM based on DSmT is proved to be a valid system, especially for fusing imprecise, uncertain and even high conflicting information and for building accurate map; the experimental results also prove the validity and superiority of the system for dynamic path planning. In the past, A* algorithm was commonly used for in global path planning only, but here it is shown that it can also be used efficiently for the local path planning as well.

ACKNOWLEDGMENT

This study is supported by the National Natural Science Foundation of China under grant No. 60675028 which is greatly acknowledged.

REFERENCES

Belkhous, S., A. Azzouz, M. Saad, C. Nerguizian and V. Nerguizian, 2005. A novel approach for mobile robot navigation with dynamic obstacles avoidance. J. Intell. Rob. Syst., 44: 187-201.

Chiang, C.H., P.J. Chiang, J.C.C. Fei and J.S. Liu, 2007. A comparative study of implementing fast marching method and a search for mobile robot path planning in grid environment: Effect of map resolution. Proceedings of the IEEE Workshop on Advanced Robotics and Its Social Impacts, December 9-11, 2007, Hsinchu, Taiwan, pp: 1-6.

Dezert, J. and F. Smarandache, 2003. On the generation of hyper-powersets for the DSmT. Proceedings of the 6th International Conference of Information Fusion, Volume 2, July 8-11, 2003, IEEE Xplore, pp: 1118-1125.

Dezert, J. and F. Smarandache, 2004. Advances and Applications of DSmT for Information Fusion (Collected Works). 1st Edn., American Research Press, Rehoboth, New Mexico, USA., ISBN: 1-931233-82-9.

Dezert, J. and F. Smarandache, 2006. Advances and Applications of DSmT for Information Fusion (Collected Works). 2nd Edn., American Research Press, Rehoboth, New Mexico, USA., ISBN: 1-59973-000-6.

Dezert, J. and F. Smarandache, 2009. Advances and Applications of DSmT for Information Fusion. Vol. 3, American Research Press, Rehoboth, ISBN: 9781599730738, Pages: 734.

Dezert, J., 2002. Foundations for a new theory of plausible and paradoxical reasoning. Inform. Secur., 91: 13-57.

Dubois, D. and H. Prade, 1986. On the unicity of Dempster rule of combination. Int. J. Intell. Syst., 1: 133-142.

Grisetti, G., G.D. Tipaldi, C. Stachniss, W. Burgard and D. Nardi, 2007. Fast and accurate SLAM with Rao-Blackwellized particle filters. Rob. Auton. Syst., 55: 30-38.

Latombe, J.C., 1991. Robot Motion Planning. Kluwer Academic Publishers, New York.

Li, X., J. Dezert and X. Huang, 2006a. Selection of sources as a prerequesite for information fusion with application to SLAM. Proceedings of the 9th International Conference on Information Fusion, July 10-13, 2006, Florence, pp: 1-8.

Li, X., X. Huang, M. Wang and J. Dezert, 2006b. A fusion Machine based on DSmT and PCR5 for robot's map reconstruction. Int. J. Inf. Acquisition, 3: 1-11.

Li, X., X. Huang, J. Dezert, Z. Wu and H. Zhang, 2007. DSmT-based generalized fusion machine for information fusion in robot map building. Proceedings of International Colloquium on Information Fusion, August 22-25, 2007, China, pp: 63-70.

Li, Z.X. and T.D. Bui, 1998. Robot path planning using fluid model. J. Intell. Rob. Syst., 21: 29-50.

Liang, Z., X. Ma, X. Dai and F. Fang, 2009. Distributed-perception-based simultaneous localization and mapping for mobile robots. Robot, 31: 33-39.

Minguez, J. and L. Montano, 2004. Nearness diagram (ND) navigation: Collision avoidance in troublesome scenarios. IEEE Trans. Rob. Autom., 20: 45-59.

Moravec, H. and A. Elfes, 1985. High resolution maps from wide angle sonar. Proceedings of IEEE International Conference on Robotics and Automation, March 25-28, 1985, IEEE Xplore, pp: 116-121.

Murphy, C.K., 2000. Combining belief functions when evidence conflicts. Decis. Support Syst., 29: 1-9.

Noykov, S. and C. Roumenin, 2007. Occupancy grids building by sonar and mobile robot. Robotics Autonomous Syst., 55: 162-175.

Shafer, G., 1976. A Mathematical Theory of Evidence. 1st Edn., Princeton University Press, Princeton, New Jersey, USA., ISBN: 9780691081755, Pages: 297.

Voorbraak, F., 1991. On the justification of Dempster's rule of combination. Artif. Intell., 48: 171-197.

Yang, X., M. Moallem and R.V. Patel, 2005. A layered goal-oriented fuzzy motion planning strategy for mobile robot navigation. IEEE Trans. Syst. Man Cybern. B, 35: 1214-1224.

Yenilmez, L. and H. Temeltas, 2007. A new approach to map building by sensor data fusion: Sequential principal component-SPC method. Int. J. Adv. Manufact. Technol., 34: 168-178.

Yin, L. and Y. Yin, 2008. An improved potential field method for mobile robot path planning in dynamic environments. Proceedings of the 7th World Congress on Intelligent Control and Automation, June 25-27, 2008, Institute of Electrical and Electronics Engineers, Chongqing, China, pp: 4837-4841.

Zadeh, L.A., 1979. On the validity of Dempster's rule of combination of evidence. University of California, Berkely.

Zadeh, L.A., 1984. Review of mathematical theory of evidence. AI Magazine, 5: 81-83.

Zadeh, L.A., 1985. A simple view of the Dempster-Shafer theory of evidence and its implications for the rule of combination. Berkeley Cognitive Science Report No. 33, University of California, Berkeley, CA.

Zadeh, L.A., 1986. A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination. AI Magazine, 7: 85-90.

Zhang, X., A.B. Rad and Y.K. Wong, 2008. A robust regression model for simultaneous localization and mapping in autonomous mobile robot. J. Intell. Rob. Syst., 53: 183-202.

Zhong, Y., B. Shirinzadeh and Y. Tian, 2008. A new neural network for robot path planning. Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, July 2-5, 2008, Xi'an, China, pp: 1361-1366.