A Hybrid Planning Approach for Robots in Search and Rescue

Sanem Sariel

Istanbul Technical University,

Computer Engineering Department

Maslak TR-34469 Istanbul, Turkey.

sariel@cs.itu.edu.tr

ABSTRACT

In this work, a hybrid planning and behavior module design is proposed for a search and rescue robot. The design was implemented based on the primary goal of the robot to find victims in the disaster area within a minimum possible time interval. For this purpose, effective exploration and exploitation algorithms are proposed. The proposed algorithms mainly provide avoiding static or dynamic obstacles for preventing further collapses, avoid cycling in the search, search the area effectively via optimum path plans, handle errors, and re-plan to cope with errors. This proposal of the hybrid planning strategy is believed to be promoting further improvements in search and rescue robot designs.

Keywords: search and rescue robot, planning and behavior, hybrid planning

1. Introduction

Because of the tiring job of Search & Rescue teams (i.e. experts, workers and dogs using some sensory information) in great disasters, the design of a Search & Rescue Robot becomes a necessity. Especially, the earthquakes we faced make clear the necessity of robust, dynamic, and intelligent planning systems and powerful human machine systems to cope with changing situations to best save people. Because the scale of the disaster and speed of changing situation is far beyond human-based mission planning [2]. There are many difficulties with human team members in disaster areas. The search operation is sometimes interrupted by some duration because of the small number of the qualified team workers. The work could be very dangerous for the rescue team members. There may be unreachable places for the rescue team members. To overcome these difficulties, the robots suit well for this job. Rescue dogs could help reducing the human risk by searching smaller voids in the rubble than a human can, but they cannot replace a video camera or structural-assessment equipment [4]. The importance of the Search & Rescue robots is emphasized by many of the researchers [1]. The research on this area has become very attractive during the last decade, and many robot architectures have been proposed and also a RoboCup-Rescue League started in 2001 [6].

There are many robot types proposed in recent works. The variety is very large from hexapot legged robots having a rod with a microphone and micro-CCD camera to robots small enough to go into the debris with CCD and infra-red camera [2]. Serpentine and shape-shifting robot types could provide very effective search operation on 3D environment with their high number of degrees of freedom [5]. Using marsupial team of robot types may also be very effective for searching the debris [4]. Since the mechanical designs are highly focused in these works, the planning strategies are not addressed in detail.

As the most important part of a SR robot, the planning layer should be capable of generate effective plans on finding victims in a short period of time while taking care of some constraints.

For the planning layer to operate successfully, a description of the world state is needed. However, in real situations, this is not always possible. The environment is partly observable, beside being dynamic and unpredictable. The actions are instantaneous and non-deterministic. The percepts are imperfect. The uncertainties about the environment and the effects of actions, the environmental constraints, goal interactions, and time and resource constraints make the problem harder. Therefore, flexible search strategies should be designed to make the problem easier. The time is a very important competitor while finding the humans suffered from the disaster. There is a tradeoff between effective planning without the dead ends and the fast plans for finding more victims in the environment.

Partial-planning or re-planning [3] strategies are known to be very effective. However these approaches are not directly suitable for real implementations of the SR robots. Therefore key points of both approaches should be used in the design of a planning layer design of a SR robot while taking the advantages of both.

There are some architecture types which can be used in planning layer design. Three layer architectures as InteRRap model and BDI (Belief-Desire-Intentions) architectures may be selected to implement in planning layer based on the applications [7].

In this study, a hybrid planning strategy is proposed for SR robots. The layered model of InteRRap and the belief update procedures of BDI type architecture were combined, and re-planning strategies were attached to the planning layer action selection mechanism to cope with the uncertainties on the environment. Reactive actions are executed directly according to the sensory information. This proposal of the hybrid planning strategy is believed to be promoting further improvements in search and rescue robot designs.

2. Proposed Hybrid Planning Module Design

Planning and Behavior Module (PBM) of a SR robot interacts with other parts of the robot to make decisions and to convert these decisions into actions. Since the other parts of the design of a search and rescue robot is out of scope of this work, the incoming information from the other modules are assumed to be fed to the PBM. Four types of information are obtained from the other modules of the robot. A probability information (P(x,y,z) in 3D environment) about locations of victims is obtained from the Locating human Beings Module of the robot. The current obstacle information from the environment (O(x,y,z)) comes directly from Obstacle Analyzer Module. The location information of the robot (R(x,y,z)) is taken from the 3D Map and Localization module. The communication interface module converts coming radio signals from the other agents in the environment into an understandable message form (such as packages of any ACL, Agent Communication Language [7]). The actions based on the plans produced by the PBM are sent to the motor interface unit of the robot. The interactions between the PBM and the other modules can be seen in Figure 1.

The PBM should be able to:

- determine a plan based on the environment's current state, the robot's location, the hierarchical structure of the desires (goals), and the constraints.
- perform re-planning, while executing a plan based on the intermediate changes on the environment or the internal state.

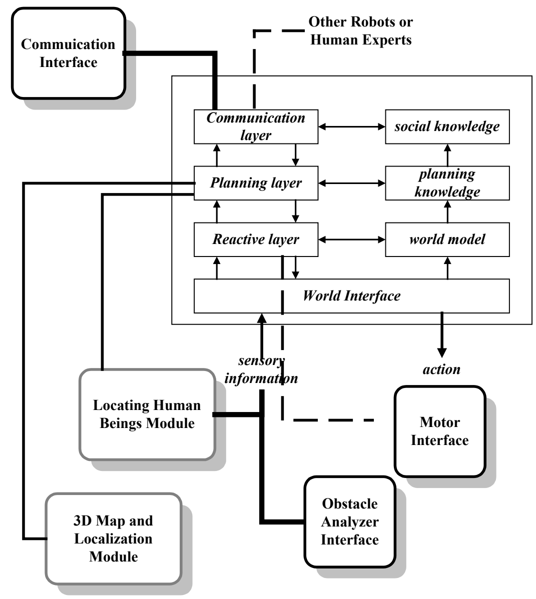


Figure 1 : The connectivity diagram of the layers of the PBM

The primary goal of the PBM is to locating victims in the disaster area within a minimum possible time interval. The sub-goals can be listed as: avoiding obstacles, avoiding risking resources or triggering a further collapse, searching the area effectively via optimum path plans, error handling, re-planning, turning and moving to the directions of locations on which it is believed that humans to be rescued are located, turning and moving to the directions of locations on which it is reported that humans to be rescued are located (information form the other robots or a dispatcher), avoiding cycling in the search. The primitive behaviors of the Behavior Generator sequencer of the Planning and Behavior module are: move (straight/back), turn(left/right), avoid obstacle, wait for dynamic object. The assemblage behaviors of the Behavior Generator sequencer of the Planning and Behavior Module are: avoid cycling, search least visited areas, find alternate path, move to victim. The communicative behaviors of the Behavior Generator sequencer of the Planning and Behavior Module are: send message, receive message, pack message, extract message.

The PBM does not need to roll-back, because of the re-planning strategy which it performs in the case of a problem. The PBM implements re-planning when it encounters a problem with the current plan. This provides not dealing with expanding trees of plans. But if re-planning is applied in every step, the robot's performance decreases. In the proposed approach re-planning is implemented where it is necessary. The layers of the PBM ensure dealing with trade-offs. When time is very important, plans with small number of steps are executed. When the robot has nothing to do better, it tries to find optimum plans that can have few steps.

There may be static or dynamic obstacles, other robots, and the human workers of the rescue team in the environment. The environment structure and the map is unknown. The robot should have short-term and long-term memories. The long term memory stores the location of obstacles observed in the environment and the victim location probabilities. Short term memory stores the necessary limited information for avoiding cycling in the search.

2.1. PBM Layer Decomposition

The Planning and Behavior Module consists mainly three layers as in InteRRap architecture. Layers interact with each other to achieve the same end. Reactive layer interacts directly with the sensory modules, and produces the output for the selected action. The planning layer constructs the plan for the robot to implement its task in an optimum way. The avoidance of the robot cycling in a deadlock position, forming beliefs to direct the search space, error handling and, re-planning situations implemented by this layer. FSA type is used for behavioral representations. The communication layer is the interaction layer of the module and implements the robot communications. It informs the other robots according to the planning layer outputs, and also receives the incoming information from the others. The communication among robots is achieved by a blackboard architecture [7]. Robot interactions are asynchoronous and non-blocking. The bandwidth requirements are very small, because the information sent contains only the location coordinates and a flag about the type of the information.

2.2. Algorithms Proposed for Effective Search and Rescue Operations

In the design of the PBM, effective algorithms for both exploration and exploitation are designed. The Search and Rescue robot works in a dynamic and unknown environment. The only information it has the loaded knowledge-base to operate, search and rescue. The short-term and long-term memories are initially contains only its knowledge-base determined in the design time. The robot initially uses its sensory inputs. If is has some information about the possibilities any victims, and their possible locations, it uses this information. If it cannot extract any information about the possibilities, it has to explore the disaster area effectively. That means, it should have a search strategy for exploring the area effectively. In this manner, the strategy provides a searching of the locations which are farthest from the visited locations. Therefore the robot can explore different spaces of the disaster area. While doing her job, she has to avoid obstacles, before collisions occur for avoiding further collapses.

The robot's Planning and Behavior Module operates on a finite state machine. The states can be classified as:

- CHOOSE_A_VICTIM: A victim selection strategy
- VICTIM_PLAN: Optimum path to the selected target
- RE_PLAN: re-planning strategy, while going towards a victim, an error occured
- RE_PLAN_X: re-planning strategy, while going towards a target, an error occured
- HELP_THE_VICTIM: rescuing operation for the victim
- MOVE_RANDOM: moving towards the selected unvsited location
- MOVE_INITIAL: After finishing the job going to the initial point
- SEND_FINISHED: Sending a finished job message to the dispatcher
- WAIT_FOR_SHORT: Waiting for a while in case of the obstacle in front can be a dynamic object (targert: victim)
- WAIT_FOR_SHORT_X: Waiting for a while in case of the obstacle in front can be a dynamic object (targert: unvisited location)
- HELP: Sending a help message to the dispatcher because of the errors, and dead-ends that the robot can not recover itself.

The State Diagram of the Planning and Behavior Module can be seen in the Figure 2. The arrows indicates the transitions among the states.

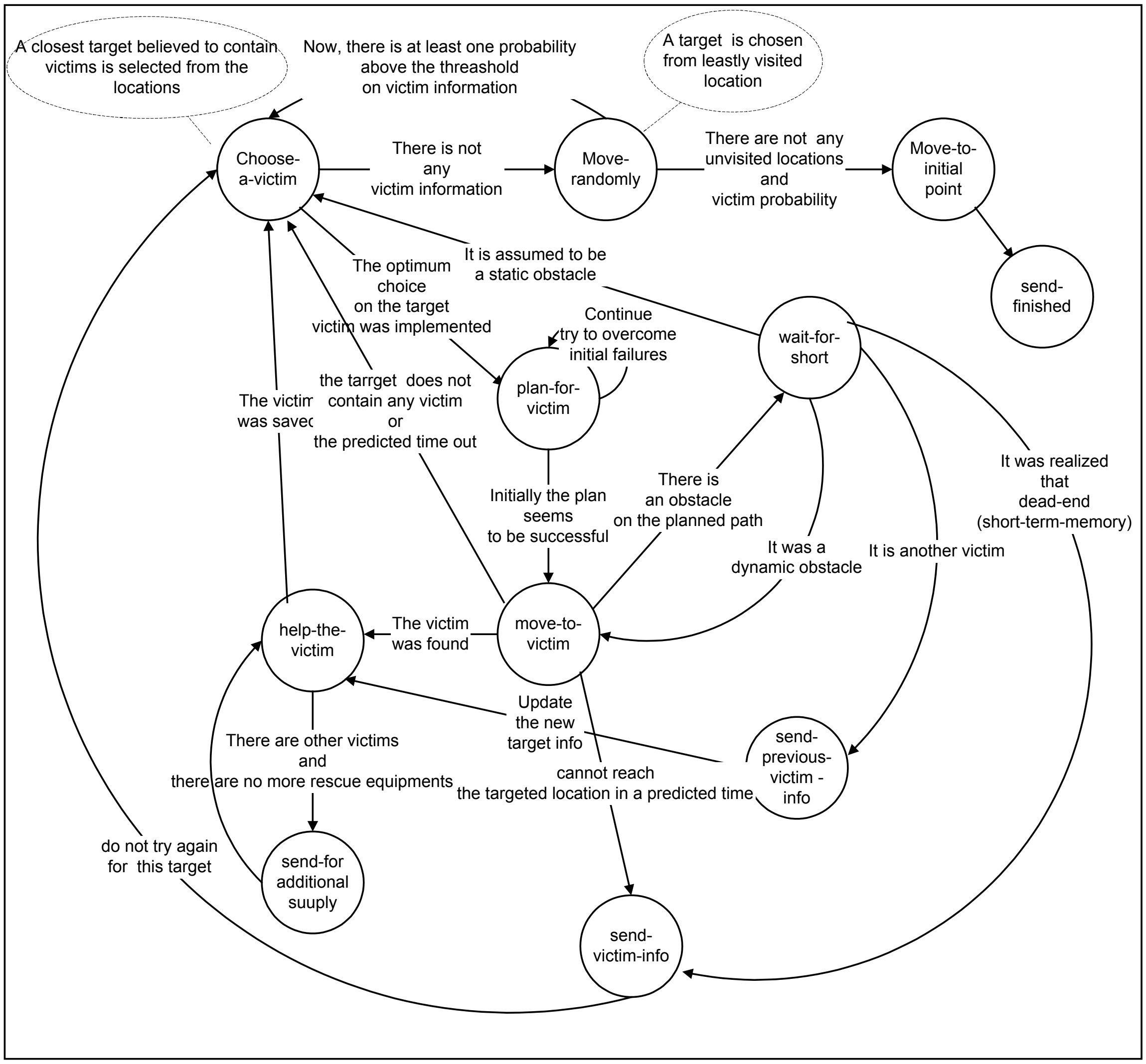


Figure 2. Finite State Machine of The Plans for the PBM

2.2.1. Victim Finding Strategy (Exploitation)

In the strategy for moving to the locations which are believed to contain one or more victims the shortest path form the robot's location to the victim location is considered. The shortest path is a direct line between these locations. Therefore the robot initially turns to the direction of the victim. Then tries to move forward continuously. The exceptions can occur when the robot finds an obstacle in front of it. It has to try other alternative directions and moving to the locations of the selected directions. After the alternative selection, the shortest victim choice is again implemented. Because the Locating Human Beings Module can indicate another victim location which is closest from the target victim. The newly arrived information can come from another robot or the dispatcer also. In this case, the target is changed, and the closest victim is chosen to rescue because the main objective is to increase the number of rescued victim in a short time. Therefore analyzing coming sensory information in every step provides the robot behave more intelligently. When the robot changes its target, it can send the skipped target information to the dipatcher. Thus other robots may rescue for this victim. The victim choosing and moving towards the selected victim algorithm can be seen in Figure 3.

Choose the closest location which is believed to contain one or more victims
Plan for the selected location
Use the location information of the robot and the location of the victim (belief)
Turn to the direction
If there is an obstacle in the next location
Wait for a while
If the obstacle is still there select a new rotation
If short-term mem. indicates that the robot cycles
If it cannot recover itself
Send a help message , idle mode
Else select different target, (choose the closest)
Move forward
If another victim is found or Locating Human Beings Module specifies another victim location which is closest to the robot, select this location as the new target, send the information of the old target to the dispatcher and continue form the second step (plan for the selected location)

If the location is reached, but there is not any victim found, clear the belief of this location and return to the first step (choosing the closest)
Move forward and return to the first step (choosing the closest)

Figure 3. The victim choosing and moving towards the selected victim algorithm

The strentgh of this algorithm is that, the robot examines the information coming from the other robots or from its motivation (a possibility from the Locating Human Beings Module) in every choosing step. Therefore the robot does not have to execute expanding trees of plans. There is another utility that the robot dynamically examines its world knowledge which is updated by the Locating Human Beings Module Interface and the Communicaiton Layer. It can select a different target in the next selection. This strategy provides the robot to traverse a path, visiting victim places by using the shortest path. Therefore the time constraint is being considered.

For overcoming the cycles in the search, a short-term memory is stored as a vector in the implementation. This short-term memory is updated when the robot encounters a problem. If a problem occurs again, the short-term memory is searched. If the same situation is stored in the short-term-memory, the robot chooses a different target. If there are no more targets or the robot cannot recover itself from the current situation it sends a help message to the dispatcher. The robot also predicts a deadline for the selected target. If this deadline is reached before going to the target, the selected target is changed, and the old target's information is sent to the dispatcher.

The robot stores the probabilities information for the locations as a vector in the long-term-memory. If the robot finds a victim in a location, it clears the probability of the current location, and saves the location as rescued locations. It sends a saved message to the dispatcher. This kind of rescued information message can arrive at the communication layer of the current robot. If the location in the message is the target location, the robot clears the probability information for this location and selects another target location in the next step.

2.2.2. Searching Unvisited Locations Strategy (Exploration)

When the robot has nothing to do better, it searches the environment effectively. This situation occurs when there is not any motivation about a probability of a location which is believed to contain a victim. This probability information can be generated either by the work of the Locating Human Beings Module or by the other robots' messages.
In "the search the area" mode, the robot should visit the unvisited locations by using an effective strategy. A searching strategy for implementing effective exploration of the disaster area is proposed. The algorithm can be seen in Figure 4.

Initially the pressure values of all of the unvisited locations is 0.
There is not any known victim possibility.
Select a node (location) having the higher pressure value for the next target
Select a rotation (from the unvisited locations) and turn to the direction of the target.
If there is an obstacle in the next location
Wait for a while
If the obstacle is still there select a new rotation
If short-term mem. indicates that the robot cycles
If it cannot recover itself
Send a help message , idle mode
Else select different target
Move forward
Update pressure values of the unvisited locations vector
Try to choose a location which is believed to contain a victim
If there is still no information on victim locations
continue from the first step(selecting a node having the higher pressure value)
Else (another victim is found or Locating Human Beings Module specifies a victim location) select this location as the new target, (plan for the selected victim), execute steps for the victim plan

Figure 4. Unvisited location selection algorithm

A selective pressure value is defined for the unvisited locations. This value for an unvisited location is updated for each moving action of the robot. The added value to the previous pressure value is proportional to the distance from the currently visited location. Therefore the robot chooses an unvisited location in each step which is the farthest from the robot's traversed path. The next chosen unvisited location is the location having the higher selective pressure value. A pressure value of a cell is updated based on Eq. 1. In this equation pr_i is the pressure value of the cell i. Dist() function is the Euclidian distance between two points in 3D space. i(x,y,z) is the cell location. R(x,y,z) is the robot location.

$$pr_i^{k+1} = pr_i^k + dist(i(x,y,z), R(x,y,z)) \quad (1)$$

While going towards the unvisited locations, the robot tries to choose the next location from the unvisited locations. When traversing the path, it removes the visited locations from the unvisited locations vector.

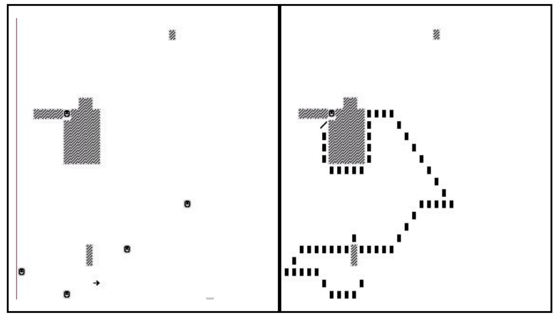
The next unvisited location selection is made based on analyzing –45, -90, 0, 45, 90 rotation locations, their visited information, and the pressure values. If all of the locations were visited, the 0 rotation location (forward direction) is chosen, if there is no obstacle there.

3. Experimental Results

To Test Proposed PBM a 2D simulation environment was generated by C++ programming language. The designed layers of the PBM was represented as different classes.

A sample trajectory path of the simulated robot's from the simulator's output can be seen in Figure 5. the arrow indicates the robot' s initial location and the direction. There are 5 victims in the environment. The locations of the dynamic and static obstacles are defined randomly in each run of simulation. The dynamic obstacles' behavior, and new locations are also defined randomly. Because of the unknown structure of the environment, the robot's selection is defined by its beliefs, and the selection directions.

There are some situations that the robot cannot recover itself, like cycling in a circle when it is surrounded by static or dynamic objects. In this case when the threshold time to move the target is reached the PBM produces an error situation.



(a) Initial view (b) Path for finding the victims

Figure 5. A sample trajectory generated by the PBM by using the proposed algorithms in 2D

4. Conclusion and Future Work

The design of the Planning and Behavior Module, directly affects the Search and Rescue Robot's success on finding the victims and rescuing them. The environment is dynamic and partially observable. The uncertainties about the environment and the effects of actions, the environmental constraints, goal interactions, and time and resource constraints make the problem harder. In this work, a hybrid planning and behavior module design is proposed. The design was implemented based on the primary goal of the robot to find victims in the disaster area within a minimum possible time interval. For this purpose, effective exploration and exploitation algorithms are proposed. The proposed algorithms provide avoiding static or dynamic obstacles for preventing further collapses, avoid cycling in the search, search the area effectively via optimum path plans, handle errors, re-plan, turn and move to the directions of locations on which it is believed that humans to be rescued are located, turn and move to the directions of locations on which it is reported that humans to be rescued are located (information form the other robots or a dispatcher), informing other robots according to beliefs or exact information about humans to be rescued.

The robot is assumed to be working in a homogeneous multi-robot team consisting of a dispatcher providing communicative operations and some forwarding operations initially. As a future work, a distributed agent team could be implemented for more effective strategies. However, this proposal of the hybrid planning strategy is believed to be promoting further improvements in search and rescue robots designs.

References

[1] Davids, A., "Urban search and rescue robots: from tragedy to technology," *Intelligent Systems, IEEE* , Vol. 17 No. 2 ,pp. 81-83 March-April 2002.

[2] Kitano, et al., "RoboCup-Rescue: Search and Rescue for Large Scale Disasters as a Domain for Multi-Agent Research," *Proceedings of IEEE Conference on Man, Systems, and Cybernetics(SMC-99)*, 1999.

[3] R. R. Murphy, "Introduction to AI Robotics", *England: The MIT Press*, 2000.

[4] R.R. Murphy , "Marsupial and shape-shifting robots for urban search and rescue", *Intelligent Systems, IEEE,* Vol. 15, No. 2 , pp. 14-19, March-April 2000.

[5] M. Sato, M. Fukaya, and T. Iwasaki , Serpentine locomotion with robotic snakes, *Control Systems Magazine, IEEE* , Vol. 22 No. 1 , pp. 64 –81, Feb. 2002

[6] T. Takahashi and S. Tadokoro, "Working with robots in disasters", *Robotics & Automation Magazine, IEEE* , Vol. 9 No. 3 , pp. 34 –39, Sept. 2002.

[7] G. Weiss, *Multi Agent Systems: A Modern Approach to Distributed Artificial Intelligence*, England: The MIT Press, 1999.