Rescue Simulation League - Virtual Robots 2011 Pasargad Team Description Paper

Yaser Abbasi[1], Fateme Hemati[1], Abbas Razzaghpanah[1], Shiva EbrahimiNejad[2], Sayyed Naser Hasani[1], Mohammad Eghlima[3], Soheil Tork Abadi[4], and Arash Nouri[1]

[1] Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran
[2] Department of Mechanical Engineering, Amirkabir University of Technology, Tehran, Iran
[3] Aptech ACCP, Tehran, Iran
[4] Information Technology Department, Southern Cross University, Australia
{yas.abbasi,fateme.hemati,r.panah,shiva.ebrahimi,arash1718}@aut.ac.ir
{snhasani,eghlima.r,soheil.torkabadi}@gmail.com

Abstract. The team started in the summer of 2010, in hope for providing enhanced coding aid for the Pasargad Real Rescue Robot team. Since, the main goal has been set to make a useful environment for implementing, as well as testing algorithms such as SLAM and obstacle avoidance algorithms. This paper is going to briefly describe the work done so far to achieve that goal.

1 Introduction

Pasargad Real Rescue Robot team have a notable reputation in robotic competitions. In Pasargad robotics club, a group of students in fields of computers, electronics and mechanics have gathered to develop robotics in various aspects. The club consists of two virtual and real rescue robotic teams that collaborate in both types of competitions, forming a joint team.
Team members and roles are as follows:

GUI Development and SLAM:	Yaser Abbasi
SLAM and Map Design:	Fateme Hemati
GUI, Communication and Exploration:	Abbas Razzaghpanah
Robot Design and Porting:	Shiva Ebrahimi Nejad
SLAM and Navigation:	Sayyed Naser Hasani
Autonomy and GUI:	Mohammad Eghlima
Autonomy and GUI:	Soheil Tork Abadi
Victim Detection:	Arash Nouri

The computer programming team in both teams are a unified group trying to import real robots in the USARSim simulation world to decrease time spent testing the real robots and thus, reducing development costs. More in [1].
So far, the virtual robot team have created a map close to what's used for real

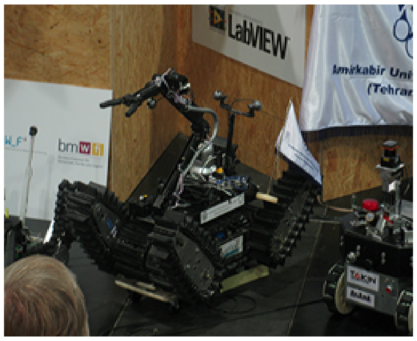

Fig. 1. Pasargad Autonomous Rescue Robot in Action

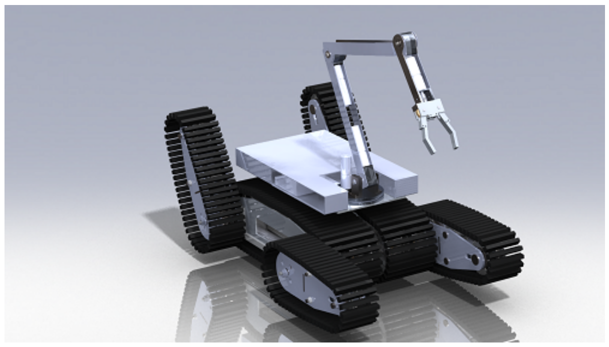

Fig. 2. Solid Model of Pasargad Autonomous Rescue Robot

robots and is currently modeling the real robot in the simulation environment in order to simplify implementation of localization, mapping and obstacle avoidance for the real robot by simulating them in a virtual world.
The team have had advances in teleoperation and is looking forward to use joypads [2] and voice control commands as a means of robot control in future events.

2 Map Design

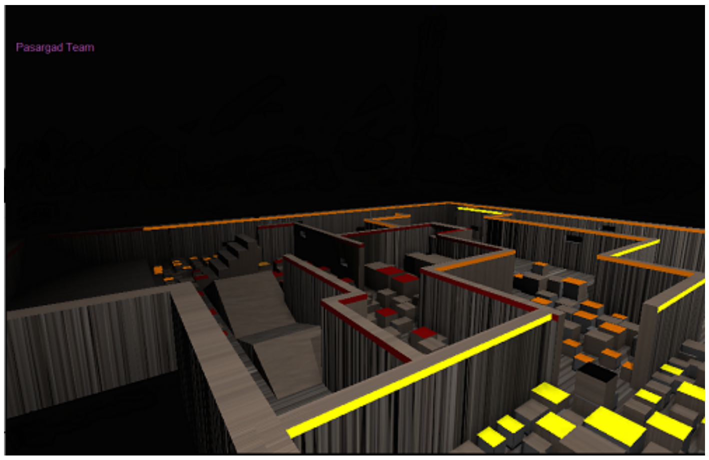

Fig. 3. Overview of a Generated Map

In order to have a better simulation of what the real robot has to go through, we need to design maps that are fairly similar to the real disaster field and standard fields used in real robot competitions.
This could also come in handy when designing a map and testing a robot in it, before building the real field.
More information about the map and download links are provided here:
`http://www.pasargadrobotic.com/pages/download.html`

3 Localization and Mapping

In today's world, robots have to operate in environments and terrains that may or may not be known. In order for a robot to navigate through such places, a simple map of geometry and obstacles should be generated and the robot's location within that map should be estimated. And this, has to be done using sensors and cameras mounted on the robot. The SLAM algorithm can produce a low-error map with the estimation of robot's placement and direction based on data streams from sensors that are updated as the robot moves. Topology-based maps can be automatically extracted from an occupancy grid built from sensor data using techniques borrowed from the image processing field. Since these techniques can be soundly defined on fuzzy values, our approach is well suited to deal with the uncertainty inherent in the sensor data. Topology-based maps are fairly robust with respect to sensor noise and to small environmental changes, and have noise computational properties. [3]

Topological mapping and metric mapping are the main approaches to obtain

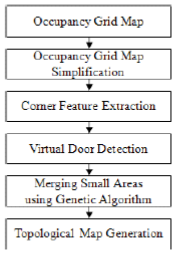
Occupancy Grid Map
Occupancy Grid Map Simplification
Corner Feature Extraction
Virtual Door Detection
Merging Small Areas using Genetic Algorithm
Topological Map Generation

Fig. 4. Mapping Steps

a map. The topological map consists of nodes that represent the significant spaces in the environment, and edges that represent the connection between the spaces. For this reason, the topological map is compact and suitable for global path planning. The topological map makes the cleaning robot simply perform room-to-room path planning by dividing the environment. A cleaning robot is one that uses maps that are created using these methods to carry out search and rescue tasks. Note that searching room to room helps finding victims better by dividing the disaster space and searching rooms with higher probability of containing victims.[4]

Virtual door is defined as all possible locations of real doors in an occupancy grid-map, and the occupancy grid-map is obtained in orthogonal environment. Virtual doors are detected by extracting corner features from occupancy grid-map, and an initial topological map is generated which has many nodes (rooms) and edges (virtual doors). The final topological map is generated by using a genetic algorithm (GA) to merge the nodes. [5]

GA is suitable for complex problems or poorly understood problems that have large search space. For example, if twenty virtual doors are detected, the problem has 220 solutions in the search space because each virtual door has two options: removed or not. In the case of exhaust search, all solutions need to be evaluated, this task has a large computational load, and is not time-efficient. GA can be used to decrease the computational load and time for the optimization of the problem. A detailed description of proposed methods are described in [5].

A robot bases it's decisions on the place in which it is located on the map. In

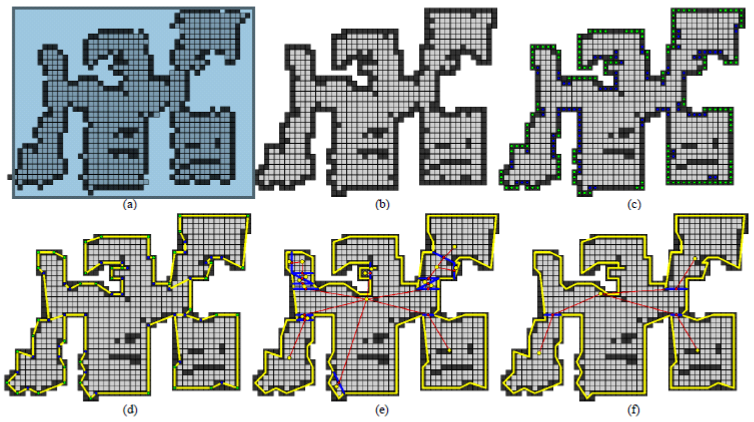

Fig. 5. Green circles: concave features; blue circles: convex features; yellow circles: nodes located in centers of rooms; red circles: edges located in centers of virtual doors; yellow line: boundary of indoor environment; blue lines: virtual doors; red lines: connections between each edge and neighboring nodes. (a) Occupancy grid-map, (b) simplified occupancy grid-map, (c) sampling of obstacle grids, (d) corner feature extraction, (e) initial topological map, (f) merged topological map.

order for the robot to localize, it uses absolute and relative measurement data of it's driving actions and a rough image of it's surroundings. The "Kalman Filters" is the algorithm used to localize the robot by considering all possible mappings and robot placements and picking the ones that are most likely the real ones. Kalman Filters algorithm is an iterative and adaptive approach to localization, meaning that technical soundness is improved by increasing the number of iterations on mapping data. [6, 7]
A fixed laser sensor mounted on the robot can produce error in mapping since the robot may misestimate distance while treading on inclined surfaces. To overcome this, a stabilizer is used to keep the laser sensor horizontally level.

4 Exploration and Victim Detection

In a simulated disaster space, a number of robots have to be controlled and driven toward finding and indicating the place of a victim, in a timely fashion and by the least possible number of human controllers. In order to achieve that, we've divided the states each robot can be in, into three interchangeable states: autonomous, semi-autonomous and manually controlled. Overall control is carried out by one human operator monitoring video feeds and mapping data of all robots. In the beginning, the human operator dispatches all robots in different directions by putting them in autonomous mode. Then the operator checks video feeds of all robots to see if there's any sign of a victim, and, if there's any, takes control of the robot and drives it next to the victim. Note that the original idea was to have a fully automatic victim detection system, but since there's no victim sensor present yet, and our project to develop a victim detection system based on machine-vision wouldn't be ready for this year's competition, we decided to use aid of a human operator in order to find as many victims as possible. Also, in light of clear indication in this year's set of rules, it's implied that autonomy is prioritized above tele-operation since the score is divided by H^2, where H is the number of human operators. In order to fulfill that, we've developed the semi autonomous system based on what we had in the base code. The semi autonomous robot can find it's way through to a chain set of points given by the human operator. Since there's a lot of robots to be controlled by a human operator, we've also come up with some ideas to enhance the user interface in many ways.

5 Connection Management and Multi-Client Control Handling

Regulating and managing control client (or clients) necessitates an expert connection manager that can supervise several streams of commands and state-data stream transmission (it could also conduct several teleoperators to communicate with the same connection). This data manager could also directly handle image data in order to preserve bandwidth for decision server.
A proxy server should be implemented to manage data flow between clients and the decision server, and another should be in charge of communication of the decision server and the robot interfaces. The latter could also come in handy as layer interfacing with the server when rules and command regulations are subject to frequent change. In addition, this method makes offline debugging and testing easier since the proxy server can simulate a given condition off a log file, as many times as desired. Offline debugging is the act of tracing the exact set of events to find the problem that's caused by those events. In order to do that, proxy server could provide a pre-recorded feed for the decision server to simulate the circumstances under which the problem procures, in order to ease debugging.
Having stated its importance, we've considered implementing two proxy servers for use with our agents. The diagram below schematically traces what we have in mind.

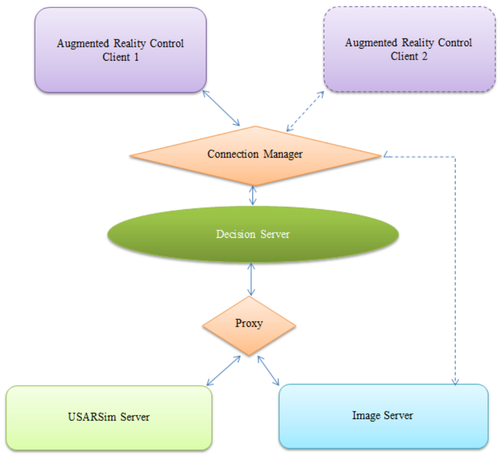
Augmented Reality Control Client 1
Augmented Reality Control Client 2
Connection Manager
Decision Server
Proxy
USARSim Server
Image Server

Fig. 6. Communication Structure

6 Software Used and Development

Passargad virtual robot team mainly uses base communication codes from SEU team's toolkit written in Java for teleoperation and may further expand and enhance it. In addition, for autonomous robots, the code for UsarUI is being considered and expanded.
So far, efforts have been made to make a user interface, similar to the real teleoperated robot's user interface.

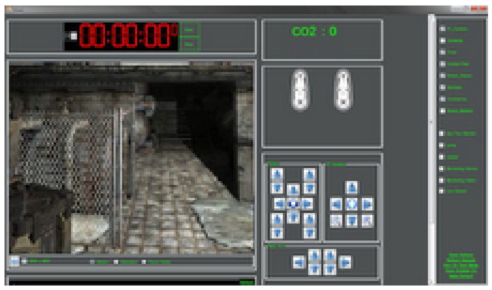

Fig. 7. User Interface From The Real Robot

7 Future Aspects and Real World Applicability

Pasargad virtual robot team would like to participate in the 2011 International Robocup Event in Istanbul and bring the collaboration between the real robot and virtual robot teams to a whole new level.

References

[1] Jijun Wang, Stephen Balakirsky, Stefano Carpin: USARSim: a robot simulator for research and education

[2] Clayton Walnum: Teach yourself Game Programming with DirectX, SAMS

[3] E. Fabrizi and A Saffiotti: Extracting Topology-Based Maps from Gridmaps, International Conference on Robotics and Automation, vol. 3, 2000, pp. 2972-2978.

[4] E. Fabrizi and A. Saffiotti: Augmenting topology-based maps with geometric information, Robotics and Autonomous Systems, vol. 40, (2002), pp. 91-97.

[5] Kwangro Joo, Tae-Kyeong Lee, Sanghoon Baek, and Se-Young Oh: Generating Topological Map from Occupancy Grid-map using Virtual Door Detection, IEEE.

[6] Professor Rick Middleton: Applications of the Kalman Filter Algorithm to Robot Localisation and World Modelling, Electrical Engineering Final Year Project, University of Newcastle, NSW, Australia, 2002.

[7] Kalman, R. E. 1960. A New Approach to Linear Filtering and Prediction Problems, Transaction of the ASMEJournal of Basic Engineering, pp. 35-45, March 1960.