

Robocup 2008 –Rescue Simulation League Virtual Robot Team Description IUST (IRAN)

Majid Sadeghi Alavijeh¹, Behzad Tabibian¹, Khoshnam Shojaee²
Soroush Mortezaipoor¹, Naser Mozaiani¹, Alireza Mohammad Shahri²

¹ IUST Robotic Association Labs, Computer Engineering Department,
Iran University of Science and Technology, Tehran, Iran
{Majid.Sadeghi.Alavijeh, btabibian, S.mortezapour, NMOzayani}@gmail.com

² Electronic Research Center, Electrical Engineering Department,
Iran University of Science and Technology, Tehran, Iran
khoshnam.shojaee@gmail.com, shahri@iust.ac.ir

<http://robotic.iust.ac.ir>

Abstract. This essay is a quick overview of design decisions made by the IUST Virtual Robot team for participating in China Robocup Competitions 2008. After an introduction to the VR League, we will give a short description of our agent architecture and the design of our Robot and techniques that will be used in order to deal with several challenging problem in Virtual Rescue Robot; SLAM (Simultaneous Localizing and Mapping) and Image Processing (in Victim Detection, Collision Detection and other similar situations). Finally, we will introduce our some parts of our future work which is based on Multi-robot SLAM Algorithm.

1 Introduction

Simulated environments are commonly used for researching artificial intelligence methods in physical multi-agent robotic systems. Simulations in Autonomous Systems are especially useful for two major problems:

- (1) To experience different situations including hazardous environments that cannot be made easily in real Environment or making these situations will cost heavily.
- (2) To study team behavior with the set of given agents.

Intention of developing and designing USARSim Environment was to build a tool for studying human-robot interaction and multi-robot coordination; it has been expanded to several other environments as well. It is also adopted as a simulation companion to the National Institute of Standards Reference Test Facility for Autonomous Mobile Robots.

Urban Search and Rescue (USAR) is the environment which we will illustrate our solutions for mentioned problems, To address the first problem we use several major algorithms such as SLAM Algorithm to localize and map the environment which the agent has got no or very few, information about it previously, this algorithm is one of

the most powerful algorithms for localizing the agent and mapping the environment simultaneously. There are some improvements in this algorithm which could be applied, such as planning and reasoning as well. For detecting victims in such Environment, we have been working on Image Processing algorithms such as Hough transform and image thresholds. We cannot use these algorithms without a tenacious architecture of the agent, later we will talk about this very important part of our research on Autonomous Mobile Robots.

Up to now we could demonstrate a powerful Architecture accompanied with several major algorithms but we cannot achieve our goals in reality unless we use several agents simultaneously which can cooperate with each other in order to accomplish the task within the shortest possible time.

2 Autonomous Navigation

We developed a traditional EKF based SLAM algorithm to meet our exploration and autonomous navigation requirements. To achieve good results in typical simulated indoor environments, we developed Line and Corner Extraction algorithms and improve them to be helpful for efficient localization and mapping process. For the sake of simplicity, we take advantage of laser range data and line extraction techniques like Split and Merge to map the environment. We are also trying to use sonar range data by applying Triangulation Based Fusion (TBF) Algorithm to extract corners and vertical lines for possible glass barriers in the environments. The nearest neighboring and JCBB techniques are also used for correct data association in our EKF-SLAM algorithm.

Unfortunately, our algorithm suffers from heavy computational cost especially in complex and messy environments. Henceforth, we are actively working on improving the algorithm to be efficient in cluttered locations by applying usual methods like sub mapping and local map joining.

The EKF-SLAM algorithm for localization and mapping, an obstacle avoidance technique, PID speed control, and combined innovative algorithms for planning and decision making will be integrated into a monolithic navigational system to meet our challenging purposes.

Another solution to minimize exploration time is using several mobile robots that they can simultaneously explore the environment. Therefore, we are trying to develop multirobot exploration techniques as well. In order to build a single accurate map of the environment, our algorithm must be able to merge all maps that are built by cooperating mobile robots.

3 Victim Detection

One of the main and challenging problems that will be considered in the urban search and rescue is victim detection. Our robot is equipped with VictSensor and camera in order to solve the mentioned problem so we can detect victim status and its

position by these equipments. Because we receive sensory data that is accompanied with some uncertainties (like white noise), we should combine sensory data with the results of processed images received from camera.

For detecting a victim in an image, we have two solutions, shape and color of the victim. The fact that victim's shape may be changed by any environmental element, force us to identify victims by the color of its skin. Therefore we define skin detection as the main problem in this approach. In order to segment human skin regions from non-skin regions based on color, we need a reliable skin color model that is adaptable to different skin colors and different lighting conditions.

In the past, we used threshold algorithm based on RGB color model that has been tuned offline. But this representation of color in images is not suitable for characterizing skin-color. In RGB space, the triple component (r, g, and b) represents not only color but also luminance. Luminance may vary across a person's face due to the ambient lighting. Therefore, it is not a reliable measure in separating skin from non-skin region. Luminance can be removed from the color representation in the chromatic color space. Chromatic colors, also known as "pure" colors are defined in the absence of luminance, which here we call it normalization process.

$$I = R + G + B \quad (1)$$

$$r = R/I \quad (2)$$

$$b = B/I \quad (3)$$

$$r + g + b = 1 \quad (4)$$

In this algorithm, the skin color distribution can be represented by Gaussian model and substituting the mean and covariance parameters of the skin color distribution in chromatic color space into the Gaussian distribution model.

In order to segment skin regions from non-skin regions based on chromatic color space, we first create a picture, based on this space properties (r, b). This picture will be a gray-scale image which gray values represent the pixels that belong to skin. All skin regions (like face, hands and arms) are shown brighter than the non-skin regions. Then we should use a threshold for determining skin sections in this picture. Because of the fact that victim's skin in real world varies, we cannot use a fixed value as the threshold, so we need an adaptive process in order to choose an optimal value for this threshold.



Fig.1 Face detection Result.

4 Locomotion

4.1 Obstacle Avoidance

Our approach to Obstacle Avoidance is based on our definition of Obstacle; in this context we define an obstacle a barrier on the way of a Robot's trajectory which prevents it from reaching its goal point. By this definition, we do not consider walls or blocks which cannot be revolved by the Robot. These barriers are parts of the map and could be seen before reaching them. In other words, obstacle is any barrier which could not be seen as a part of the map prior to reach it. In fact, all walls are parts of the map. In this definition, our Agent can revolve any obstacles in order to reach its goal position.

4.1.1 Bug Approach and its problems

We used this approach as our first Solution to avoid any kind of Obstacle. This algorithm works very well in many cases as it follows the object's contour but departs as soon as it is able to move directly toward the goal. One of the major weaknesses of this algorithm is the fact that this algorithm does not take into account robot kinematics, which can be especially important in nonholonomic robots. The other shortcoming with this algorithm is that while it uses only the latest sensor values, noises on the sensors can have serious impacts on real performance of the Robot.

4.1.2 Vector Field Histogram

As mentioned before, the major criticism of Bug Algorithm is that it is only based on the most recent values of sensors which can differ from result in real world because of sensor's noises. VFH+ algorithm which was developed by Borenstein, together with Ulrich is an extension to VFH algorithm. We used this latest algorithm because VFH is criticized of being unstable.

VFH overcomes the problem of BUG type algorithm by making a local map of the environment around the robot, which is a small occupancy grid around the robot. Data of this grid populated with the relatively recent sensors range readings. VFH produces a polar histogram. X axis of this histogram represents Alpha value at which the obstacle was found, and y axis represents the probability P, that there really is an obstacle, all based on the occupancy grid.

From this histogram, we calculate a cost function which in turn gives us the steering direction. We can demonstrate this cost function as below:

$$\text{Cost} = a. \text{Target_direction} + b. \text{Wheel_orientation} + c. \text{Previous_direction}. \quad (5)$$

Table 1. Input variables

Variable	Definition
Target_direction	alignment of the Robot path with the Goal point
Wheel_orientation	Difference between the new orientation and current wheel orientation.
Previous_direction	Deference between previously selected direction and the new direction.

By this dynamic cost function, we can adapt our Robot with a variety of limitations ranging from Kinematics limitations of Robot to the sensor noises. We should just tune this cost function against or Robot Nature and physical limitations.

4.2 Mobility

According to Planning and Victim Detection algorithms; it is required to implement a system which is capable of moving Robot from its position to a desired position in a 3 DOF Environment. There are several approaches for this problem, such as linear and non-linear Controllers or Heuristic Algorithms. However, we adopt Fuzzy controller as our main approach to solve this problem. In this context, our inputs are robots present position and desired robot position and the output of this controller is the speed of wheels that will be applied to each right and left motor in a differential drive robot.

The main idea behind our approach is that we first rotate the robot toward desired position, so we can reach that point with only moving straight toward the goal point and finally when the robot reaches its goal point it will be rotated to the desired direction. Under some circumstances Robot rotation is higher than a normal value, in such situations we attempt to move the robot in an efficient way; in this structure our input data consist of following values:

Table 2. Input variables of the controller

Variable	Definition	Formula
Distance:	Robot's distance from the goal point.	Target_Location - Curent_Location
Robot Final Direction:	Difference of current Robot's direction and the desired direction at the goal point	FinalDirection-RobotDirection
Robot Final Angel:	Angel between current Robot's Direction and the final Position	

5 Agent Architecture

Agent Architecture is the frame of all algorithms used in this paper. So without strong agent architecture, we cannot develop a stable system in which it could perform its tasks as good as possible. On the basis of this introduction, our agent architecture is fortified against any possible damages or flaws by making an Object Oriented agent architecture with multi thread system in which each thread can perform its task with any interruption from another thread also it makes it possible to get much out of new Dual and multi Core Computer Systems. In this architecture each thread passes its tasks to another thread by handling events which were defined in the class of the destination thread. As an example each Sensor's update will be carried out by an event from the world model (which in turn it will inform other parts of this architecture). This solution helps us to insure the stability of the system and prevent any possible crashes. all incoming and outgoing messages are parsed via one frame so all connections between the server and the agent will be handled by one class so any changes in server configuration can be handled easily by only changing this frame. Here we demonstrate our Sensor architecture by a UML graph of our sensor's hierarchy:

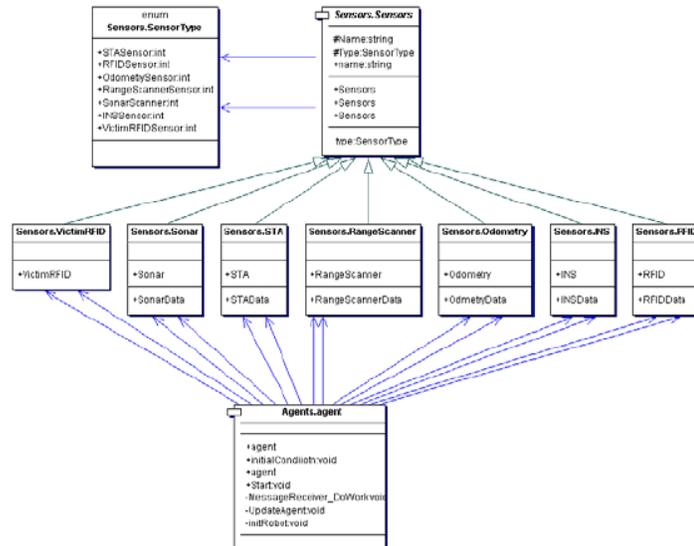


Fig.2 Sensor architecture in an UML graph.

6 Communications

In many dangerous missions and situations, it might be desirable to use multiple cost-effective robots because even if one or some of the team members is/are lost due to some failures, the others can continue to operate without any serious problem. This leads to fault tolerance and robustness. Therefore, to achieve this goal we use a way to share robot's earned information with other teammates and that is what is called exchanging information between robots. Exchanging information between robots, distributed over an area, is now possible by means of off-the-shelf ad hoc wireless network devices on real robots and that is what is simulated in VR server for robots to make communications between each other and share their earned information from environment to help the whole team to control the situation. The purpose of the Wireless Communications Server (exchanging information between robots mean) is to act as a middle man for messages passed between the robots and dropping messages and connections between robots when not possible to connect through wireless communication. The server listens on a port for connections from robots sending command messages for registering, listening and opening connections. Once connections between robots are set up, these are handled on different sockets (TcpLinks), allowing the server to listen for more commands, and allowing multiple connections to be handled. Then robots can send command for communication to server and be connected to other ones. The opening of a connection or the closing of a connection while sending a message is decided using the path loss estimated using the Wall Attenuation Factor.

$$f(d)[dBm] = f(d_0)[dBm] - 10n \cdot \log_{10}\left(\frac{d}{d_0}\right) - \begin{cases} C * WAF, & nW \geq C \\ nW * WAF, & nW < C \end{cases} \quad (6)$$

The default value for $f(d_0)$ is -49.67 dBm, d_0 is 2 m, n is 1.09, C is 5 and WAF is 6.625.

7 Future works

Our future work is mainly based on possible developments of our current architecture.

Multi-agent systems are one of the most practical approaches in Urban Search and Rescue Environments. however, We will try to use Multi-robot SLAM algorithm in order to merge the maps produced by several Robots, the way which we transmit these data between Robots are based on a Wireless medium which is carried out by a

wireless kit on each Robot, another way which we can carry out this problem is by using 1 or more UVA robot(s) which is capable of freely fling over all Robots and easily share these data between all Robots, this approach can be used mainly in outdoor environments.

One of the other developments we will do in our architecture is to improve our feature extraction by using image processing methods and algorithms.

However these are only some parts of our next approaches to improve the current architecture and we are working so hard to improve our current design.

References

1. B. Slamet and M. Pfingsthorn, ManifoldSLAM: a Multi-Agent Simultaneous Localization and Mapping System for the RoboCup Rescue Virtual Robots Competition, Master's thesis, Universiteit van Amsterdam, December 2006.
2. USARSim V3.1.1 A Game-based Simulation of mobile robots , Prepared by Jijun Wang Edited by Stephen Balakirsky
3. Roland Siegwart, Illah R.Nourbakhsh, "Introduction to Autonomous Mobile Robots," Massachusetts Institute of Technology, Prentice-Hall . 2004
4. Guido Zunino, "Simultaneous Localization and Mapping for Navigation in Realistic Environments," Licentiate Thesis, University of Stockholm, February . 2002
5. E. A. Wan and R. van der Merwe. "The unscented kalman filter for nonlinear estimation," in Symposium 2000 on Adaptive Systems for Signal Processing, 2000.
6. Rudy Negenborn. "Robot Localization and Kalman Filters on finding your position in a noisy world," M.S thesis, Utrecht University, September 2003.
7. Y. Gong and M. Sakauchi, "Detection of regions matching specified chromatic features", Computer Vision and Image Understanding, vol. 61, no. 2, 1995, pp 263 - 269