Team Description Paper
SBCe_Saviour
Virtual Robots Competition
Rescue Simulation League
Robocup 2011

Iman Sarrafi[1], Mohammad Hossein Sedighi Gilani[1], Davood Qorashi[1], Sam Serpoush[1], Ehsan Valizadeh[1], Amir Hossein Abdi[1], Eslam Nazemi[1]

[1] Shahid Beheshti University, Tehran, Iran
http://www.sbcee.net

Abstract. This paper describes SBCe_Saviour virtual robots team. This team has started its activity since 2009 and has participated in four competitions so far. In addition to the basic infrastructure, a segment-based geometric algorithm for online localization and mapping of robots and a vector based navigation system have been designed and implemented. Beside developing the SLAM algorithm, image processing for victim detection and a more advanced autonomous navigation strategy are now in the study and implementation phase.

1 Introduction

The robocup virtual robots league has provided a close-to-reality simulation platform for putting multiple scientific fields of robotics and artificial intelligence into practice. Research topics such as human-robot interfaces, autonomous navigation of robots, sensor fusion, localization and mapping, distributed planning and learning are all covered in this area.
This paper introduces the strategy of SBCe_Saviour team and explains how these challenges are handled in the team. The rest of this paper is organized as follows. Section 2 describes our localization and mapping technique. The robots' path planning and navigation strategy is described in section 3. Section 4 is dedicated to image processing and victim detection. Sections 5 and 6 explain our communication system and the operator graphical interface respectively. Finally, the conclusion is presented in section 7.

2 SLAM

In our team, a geometric SLAM algorithm suitable for online localization and mapping of robots is used due to its low processing requirements. Our main focus is on the localization problem, though precise localization will automatically lead to a precise mapping. The presented approach uses segment-based mapping [1, 2, 3], therefore it is possible to employ in multi-robot [4, 5] environments due to its lowbandwidth network requirement for transferring small map segments to a central system.

The location of a robot and the map of its surrounding environment are of great importance to robots operator and the autonomous robots themselves. We use a segment-based geometric algorithm for online localization and mapping of robots which is easily applicable in multi-robot environments. Our algorithm consists of four steps which are repeated periodically. Figure 1 depicts these steps.

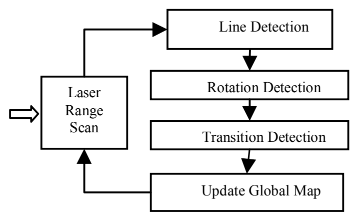


Figure 1. Main steps of the SLAM algorithm.

The following subsections describe these steps in details.

2.1. Line Detection

In order to detect lines out of the points returned by the laser scanner, starting with the first point we iterate on the array of obtained distances and test whether we can create a line with the current and previous points based on a number of thresholds. If so, the current point is appended to the list of selected points and iteration continues. Otherwise we use a line creation algorithm to create a line using the selected points, add it to the set of lines of current map, and reset the list of selected points. The iteration continues until the whole laser range data is processed. The computational complexity of this step is $O(n)$ where n is the number of points. Hence, it is a suitable algorithm for real-time line detection.

Our complete line detection algorithm is presented in detail in [6]. To better improve the line detection, based on non-crisp nature of the problem of detecting a line out a set of noisy data, we used a fuzzy algorithm [7]. This fuzzy decision making helps better line detection in noisy environments.

2.2. Rotation Detection

The second step in our method is "Rotation Detection". In this step, we try to detect the angle difference between the current generated local map segment and the previous one. In order to do this, we compare all the lines detected in the first step by the detected lines of previous map. Each couple of lines has an angle difference $\Delta\theta$ which will be normalized between -90 and 90. Considering the average length of two lines a weight will be assigned to $\Delta\theta$. This weight is assigned so that the similar lines from two consequent scans have a stronger effect on the calculated weight.
Afterwards, an array of calculated angle differences will be formed which shows the repetition of them considering their assigned weight. Since two consequent maps have

many similar lines due to the short time interval between scanning them, the rotation angle is repeated more than other angles and the weight of this angle will be amplified. Therefore, the angle with the highest weight is the most valuable repeated difference and will be chosen as the rotation angle. The result of rotation detection is used to update current direction of the robot.

2.3. Transition Detection

In this step, the rotated local map segment, which is the output of the rotation detection step, is shifted in order to match the previous maps. It is performed by comparing the current rotated map with the previous one. As it is mentioned in last section, two consequent maps are so similar due to the short time between scanning them. Therefore, there are many common lines in both maps which can be used to find the transition vector. In each map, we find the virtual joints which can be created by crossing non-parallel line pairs. These two joint sets are multiplied together creating the candidate transition vectors. Afterwards, all the vectors shift the current map and a fitness value is calculated for each transition vector. The vector with highest fitness value will be chosen as transition vector.

To demonstrate transition detection operation, suppose that the maps shown in figure 2-a are two consequent rotated maps which should be matched. The first map has one joint and the second map has three joints. Therefore, three (=3×1) vectors are created by these points and a fitness value will be calculated after shifting the second map with each of the vectors. Among the candidate vectors in figure 2-b, V_2 is the vector with the highest fitness value and will be chosen as the result.

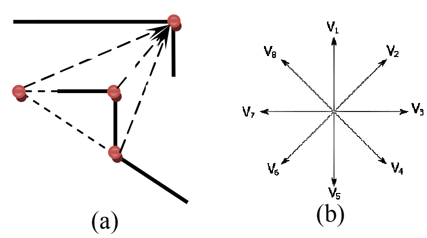


Figure 2. (a) Transition detection operation (b) Adjustment Vectors.

After finding the best transition vector, an adjustment algorithm is utilized in order to make our transition more precise. The result of the transition detection is used to update the current location of the robot.

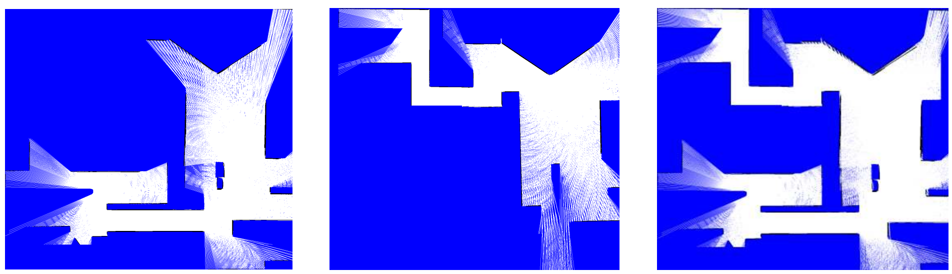
2.4. Updating the Global Map

In order to have a global map of the robot's environment, the rotated and shifted maps which are in global coordination will be appended to a global map after each iteration of the presented algorithm. Although this global map has many repeated lines, it is comprehensible enough to be used as an exploration map of the environment by both autonomous robots and human operators.

As every local map segment is in global coordination, this method can be used in multi-robot environments, where all the robots send their acquired local map segments to a central system and the appending step of map segments is done there.

2.5. SLAM Results

In this section, we provide an example of the results obtained by our method in an unknown environment. We manually moved two robots with same settings on two different routes in order to evaluate the proposed method in multi-robot situations. The results of these two robots' map building is depicted in figures 3-a and 3-b. Figure 3-c shows the result of merging these two maps on the central system.



(a) (b) (c)

Figure 3. (a) The map generated by robot 1 (b) The map generated by robot 2 (c) merged map.

3 Path Planning and Navigation

For autonomous and semi-autonomous navigation of robots, we take advantage of a method similar to the potential field [8]. To provide our navigation algorithm, we first define the following forces and rules:

1. Each line segment (wall or obstacle) has a repelling force. The repelling force is much stronger within a threshold distance in order to avoid hitting obstacles.
2. If there is a goal point or area defined by the operator, the goal effectively attracts the robot.
3. Robots repel each other. This repulsion force is also stronger within a threshold distance.
4. Traversable but uncleared (not victim-free) areas attract robots.
5. Probable or definite victim locations which could be received from different sources (detected by the image processing module, reported by other robots, or defined by the operator) are attracting forces. After getting close enough to an identified victim the robot automatically stops and remains still.
6. Every 15 seconds, the robot's current position becomes a permanent repelling force. This helps avoiding the local minima defect of the potential field algorithms.

7. An accessibility value is calculated using the wireless signal strength and the placements of the robots. The impact of the wireless signal strength must be so that it helps the robots maintain their connectivity during their exploration. On the other hand, it should not be so strong that prohibits the robots from getting separated, exploring new areas and reaching the victims.

Each of the factors introduced above acts as an attractive or repulsive force vector on the robot's position. The superposition of all these forces is applied to the robot which smoothly guides the robot toward its goals while simultaneously avoiding known obstacles.

3.1 Attractive Forces

Our attraction potential force is defined as:

$$\begin{aligned} F_{att}(q) &= -\nabla U_{att}(q) \\ &= -k_{att} \,.\, \rho_{goal}(q)\, \nabla \rho_{goal}(q) \\ &= -k_{att} \,.\, (q - q_{goal}) \end{aligned} \tag{1}$$

where k_{att} is a positive scaling factor and ρ_{goal} denotes the Euclidean distance $\|q-q_{goal}\|$. This function converges linearly toward 0 as the robot reaches the goal.

3.2 Repulsive Forces

The repulsive potential should be very strong when the robot is close to the object, but should not influence its movement when the robot is far from it. Our repulsive force is defined as

$$F_{rep} = \begin{cases} k_{rep}\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)\frac{1}{\rho(q)^2}\frac{q-q_{obstacle}}{\rho(q)} & if\ \rho(q) \le \rho_0 \\ 0 & if\ \rho(q) \ge \rho_0 \end{cases} \tag{2}$$

where k_{rep} is again a scaling factor, $\rho(q)$ is the minimal distance from q to the object and ρ_0 is the distance of influence of the object. The repulsive potential force function F_{rep} grows as q gets closer to the object.

3.3 Further Developments

In addition to better adjustment of our navigation algorithm, the extended potential field method [9] is in our perspective for future work in this segment. The rotational potential field could be utilized which assumes that the direction of the robot should influence the repulsive force values. In other words, the repulsive force should be less when an obstacle is parallel to the robot's direction of travel, since such an object is not an immediate threat to robot's movement. This method could result in better wall following.

4 Automated Human Victim Detection

Since the victim sensor is no longer supported in the competitions, we have started the study and implementation of image processing techniques for automatic victim detection. In order to detect a human victim two parameters are under focus: the geometric shape of body parts and the human skin color. The strategy is to combine the results of these two independent tasks to increase the efficiency, and decrease the false positive rate in human body detection.

In the procedure of body part detection, rectangular candidates containing each body part are selected. Then the strongest candidates are combined into a proper geometrical configuration in order to form a complete or partial view of a human body. The detection of four body parts have been included due to their unique shapes: Arm, Leg and Face and Upper body (including chest and shoulders). For the training and detecting phases the Haar classifiers of OpenCV framework are being used. This framework implements the well-known algorithm of Viola-Jones detector [10]. OpenCV has already trained an acceptable Haar classifier for face detection which we are taking advantage of it. Besides, we are training classifiers for the other three body parts mentioned above, using victim photos of the USARSim environment.

To improve the efficiency, skin color detection is performed prior to the body detection. We plan to go through a training phase where 3-channel color histograms of different samples are obtained. Then the Bhattacharyya distance coefficient is used to compute the similarity between histograms of training images and those of test regions to decide whether a region contains skin or not. Finally we restrict the face and hand detection to the areas containing skin pixels.

Figure 4 depicts how our algorithm has detected a victim's face in a test run on the UT2004 version of USARSim.

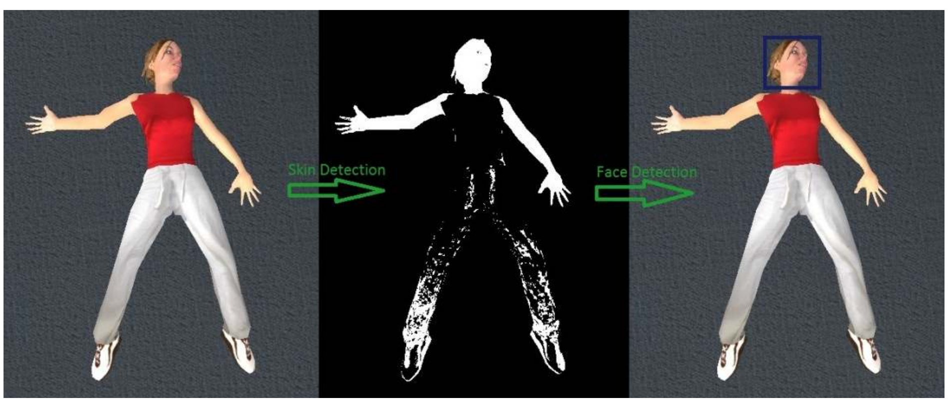


Figure 4. Victim skin and face detection

5 Communication

In addition to the general functions of the communication module, a routing layer has been included for two main reasons:

- **I.** Finding the best communication path, this is considered the path with least possible noises and intermediate agents.
- **II.** The need to a build a connected graph which provides a communication path between any two robots.

To achieve the above goals, a shortest path algorithm [12] is implemented. In this system after the communication between each two robots is started, each robot sends its signal strength with other robots to the other side in form of a package. This information is used by both sides to update their communication graphs. This graph helps robots select best possible communication paths. Each agent along such a path will act as a bridge that only forwards the received data [13]. With this mechanism robots will be able to explore far areas while keeping their communication with each other and the communication center.

6 Operator Interface

The user interface designed for communication center allows the operator to view multiple robots' mapping data, sensor data, camera images, battery status and their current locations. Via this interface the operator can control the behavior of robots by giving them direct movement orders or sets of paths to follow. The robot camera view helps the operator get a better image of the environment and control robots more efficiently; also important camera shots such as victim photos can be saved by the operator interface. Figure 5 shows the designed operator interface.

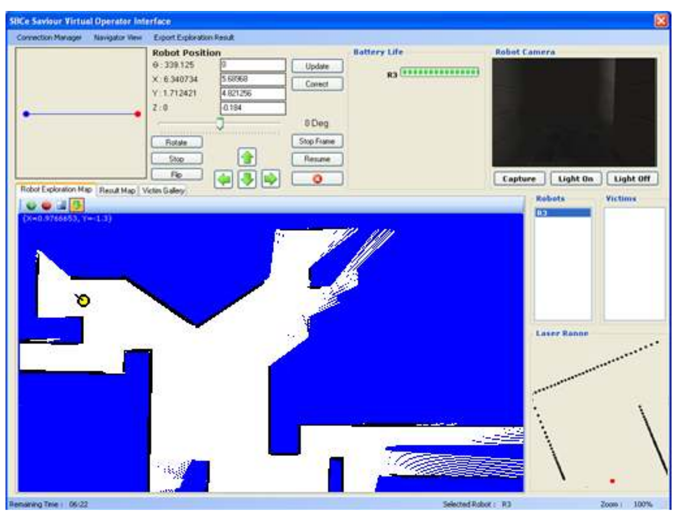


Figure 5. Operator Interface

7 Conclusion

We use an on-line geometric algorithm for solving localization problem of mobile robots. The presented approach is reliable in noisy environments and can be efficiently used in multi-robot environments.

We also take advantage of an autonomous navigation algorithm based on the potential field method. A multi-layer communication module has been developed which includes routing algorithms and supports routed communication among the team of robots. For automatic victim detection via image processing, our strategy is to combine body detection and skin color detection. Further development of SLAM, autonomy and our image processing algorithm are now in center of team's attention. Also there are changes required for better adaption to 2011 rules. We are looking forward to achieving remarkable results in the upcoming 2011 competitions.

8 References

[1] Austin, D.J. and McCarragher, B.J., Geometric constraint identification and mapping for mobile robots. Robotics and Autonomous Systems. v35 i2. 59-76.

[2] E. Zalama, G. Candela, J.Gomez, and S. Thrun. Concurrent mapping and localization for mobile robots with segmented local maps. In Proc. Of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, pages 546-551, 2002.

[3] L.J. Latecki, R. Lakaemper, X. Sun, and D. Wolter. Building polygonal maps from laser range data. In Int'l Cognitive Robotics Workshop,2004.

[4] Dieter Fox, Distributed Multi-Robot Exploration and Mapping, Proceedings of the 2nd Canadian conference on Computer and Robot Vision, p.15-xv, 2005.

[5] Parker, L., Current state of the art in distributed robot systems. In: Parker, L., Bekey, G., Barhen, J. (Eds.), Distributed Autonomous Robotic Systems, vol. 4. Springer. pp. 3-12.

[6] A. Radmand, R. Bakhshi, E. Nazemi,: An on-line geometric algorithm for localization of mobile robots using a segment-based mapping, Intl' Conference on Automation, Robotics, and Control Systems (ARCS-10), July 2010), 1-7.

[7] A. Radmandm, R.Bakhshi, E.Nazemi: Line detection from range scanner using a sequential fuzzy decision making, Robocup IranOpen 2010 Symposium.

[8] Roland Siegwart, Illah R. Nourbakhsh: Introduction to autonomous mobile robots, The MIT Press, 2004.

[9] Khatib, M., Chatila, R.: An Extended Potential Field Approach for Mobile Robot Sensor-Based Motions. In: International Conference on Intelligent Autonomous Systems, 1995.

[10] Bhaskar Chakraborty, Marco Pedersoli, Jordi Gonzàlez: View-Invariant Human Action Detection Using Component-Wise HMM of Body Parts, 2008.

[11] A. Conci, E Nunes, J. J. Pantrigo, A. Sanchez: Comparing color and texture-based algorithms for human skin detection, 2008.

[12] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to algorithms (2nd ed.) Mc Graw Hill., 2001.

[13] Doyle, J.: Routing TCP/IP Volume I (1st ed.) Cisco Press., 1998.