Abstract. With the progress made in active exploration, the robots of the UvA Rescue Team are capable of making deliberative decisions about the frontiers to be explored. The robots select the frontiers with maximum information gain. The robots incorporate the positions of their team mates in their decisions, to optimize the gain for the team as a whole. The active exploration is based on the shared occupancy map, which is online generated. During the exploration the robots are constantly looking for victims, which for the UvA Rescue Team is performed by looking for typical color histograms in the images of the omnidirectional camera.

Introduction

The RoboCup Rescue competitions provide a benchmark for evaluating robot platforms for their usability in disaster mitigation. Research groups should demonstrate their ability to deploy a team of robots that explore a devastated area and locate victims. Farsighted goals include the capability to identify hazards, provide structural support and more. Since its introduction in 2001, the RoboCup Rescue consist of two leagues; the Rescue Robot League and the Rescue Simulation League. Whereas the Rescue Robot League fosters the development of high-mobility platforms with adequate sensing capabilities, e.g. to identify human bodies under harsh conditions, the Rescue Simulation League promotes research in planning, learning, and information exchange in an inherently distributed rescue effort [1].

The Virtual Robots competition, part of the Rescue Simulation League, is a platform to experiment with multi-robot algorithms for robot systems with advanced sensory and mobility capabilities. The developed algorithms should be able to be directly used in fieldable systems, as demonstrated in several of the participating teams [2].
1 Team Members

In Suzhou 2008 the Universiteit van Amsterdam will only be represented with a single staff member, who will be active in the Organizing Committee. Fortunately, the source-code contains contributions of several master-students, which allow to operate UsarCommander fully autonomously.

UsarCommander was originally designed and build by Bayu Slamet. All other contributions are build on top of the framework of his hand. Here a list is given of the contributions previously made:

**Arnoud Visser** : supervision [2], exploration & navigation algorithms [3, 4]

**Bayu Slamet** : GUI, real time visualization [5], several scan matching algorithms, manifold-SLAM [6, 7], communication protocol [4], exploration behaviors [3]

**Max Pfingsthorn** : off-line rendering, several scan matching algorithms, manifold-SLAM, navigation behaviors [6, 7]

**Tijn Schmits** : image processing, victim detection [5], sensor development, GUI, communication protocol

**Xingrui-Ji et al.** : occupancy grid map interpretation, beyond frontier exploration [8]

**Rik Tobi, Folkert Huizinga et al.** : obstacle avoidance behaviors.

**Aksel Ethembabaoglu** : image processing, active target tracking [9]

2 Scan Matching

The possibilities for active exploration are heavenly dependend on a correct estimation of a map of the environment. A map can be generated with advanced SLAM algorithms (see for an overview [10]), but the performance of those algorithms are dependent on the accuracy of the displacement of the robot.

As laser range scanners provide rather accurate measurements of a robot’s physical surroundings they are can be used for accurate displacement estimation. *Scan matchers* are algorithms that estimate displacement by comparing the current laser scan with one or more previously acquired laser scans. Scan matching algorithms usually indeed yield more accurate motion estimates than estimates provided by odometry or inertial navigation systems. Still, also scan matching algorithm do not resolve the fact that all robot observations remain correlated through the robot’s trajectory. This implies that every estimate builds on the previous one. Hence, the estimates may still diverge incrementally from the robot’s true motions in the long run.

This fundamental issue lies at the core of simultaneous localization and mapping (SLAM) research. As such, many advanced techniques that aim to detect and correct error accumulation have been put forward by SLAM researchers. Although these SLAM techniques have proved to be very effective in achieving their objective, they usually only come into action after the error already
accumulated. With a robust scan matching algorithm the localization error is minimal, and the effort to detect and correct errors can be reduced to a minimum (see e.g. [11]).

An important part of the thesis of Slamet and Pfingsthorn [6] was already dedicated to an extensive survey of the performance of three scan matching algorithms in different environments. The original survey showed good performance indoors, and less reliable results outside. In 2007 the scan matching algorithms were extensively tested for outdoor environments. Outside there can be large free spaces, where only sparsely obstacles are detected. It was demonstrated that the robustness of the scan matching algorithms could be improved by matching against accumulated scans. With a storage technique like quad trees this accumulation can be done without losing the accuracy of the measurements.

One of the experiments was performed in the outdoor area of the 2006 Virtual Robot competition, which we call ‘The Park’ (see Fig. 1). In our experiments we will use two implementations of the ICP algorithm [12]; IDC [13] and WSM [14]. The point-correlation procedures of the original implementations were replaced with a nearest neighbor-search in a quad tree. No additional modifications were made to the internal workings of these scan matchers, so we refer the interested reader to prior research [7, 6] and the original papers for further details. The experiments will investigate the improvements that can be gained from using quad trees for both algorithms. The visualizations were created with the standard occupancy rendering techniques from [7]. All presented results are strictly based on scan matching.

The experiment covers an area of approximately 80 by 40 meters where the robot started in the bottom-right corner and traverses the park in clockwise
Fig. 2. Comparison of scan matching algorithms for a drive through a park, with poor odometry and sparse range scans.

direction. The robot’s path is shaded with gray for clarity and should describe a single closed loop from tip to tail. Both original scan matchers accumulate significant error, IDC ‘overshoots’ the end of the loop and WSM leaves a gap of several meters. Using the accumulated scans in the q-tree both IDC and WSM close the loop implicitly. Over the whole dataset the average correlation distance reduces from 9.83 mm to 4.83 mm for IDC and from 10.20 mm to 5.62 mm for WSM.

3 Localization and Mapping

The mapping algorithm of the UvA Rescue team is based on the manifold approach [11]. Globally, the manifold relies on a graph structure that grows with the amount of explored area. Nodes are added to the graph to represent local properties of newly explored areas. Links represent navigable paths from one node to the next.

The mapping algorithm is not dependent on information about the movement of the robot for the creation of links. A good estimate of the displacements can be derived only from scan matching. In practice the displacement as reported by the inertial navigation sensor is valuable as initial estimate for the scan matching. The displacement is estimated by comparing the current laser scan with laser
scans recorded shortly before, stored in nearby nodes of the graph. In principle the scan matcher can also perform a comparison with measurements elsewhere in the graph, but such a comparison is only made under specific circumstances (for instance during loop closure, as illustrated in [7]). At the moment that the displacement becomes so large that the confidence in the match between the current scan and the previous scan drops, a new node is created to store the scan and a new link is created with the displacement. A new part of the map is learned.

As long as the confidence is high enough, the information on the map is sufficient and no further learning is needed. The map is just used to get an accurate estimate of the current location. The localization algorithm maintains a single hypothesis about where the robot currently is and does an iterative search around that location when new measurement data arrives. For each point the correspondence between the current measurement data and the previous measurement data is calculated. The point with the best correspondence is selected as the center of a new iterative search, until the search converges. Important here is the measure for the correspondence. The UvA has several scan matching algorithms available (as introduced in the previous section) which can be used as correspondence measure.

The graph structure allows to have multiple disconnected maps in memory. In the context of SLAM for multiple robots, this allows to communicate the graphs and to have one disconnected map for each robot. Additionally, it is possible to start a new disconnected map when a robot looses track of its location, for example after falling down stairs.

The graph structure of the manifold can be easily converted into occupancy grids with standard rendering techniques, as demonstrated in figure 2 and [7].

4 Autonomous Multi-Robot Exploration

Exploration addresses the challenge of directing a robot through an environment such that its knowledge about the environment is maximized. A mobile robot typically maintains its knowledge about the external world in a map \( m \). Increasing the knowledge represented by \( m \) is achieved by either reducing the uncertainty about current information, or by inserting new information. The latter occurs when the map coverage is extended as the robot visits areas in the external world not yet covered by \( m \) before.

The approach in previous years [15, 5] was to passively acquire the information to store in the map, i.e. while the robot or operator was wandering around pursuing other objectives like finding victims. In this year however, the focus is on active exploration: to explicitly plan the next exploration action \( a \) which will increase the knowledge about the world the most. In this paradigm victim finding becomes the side-effect of efficient exploration.

A key inside was that the information gain for location not yet visited by the robot could be estimated from the laser range measurements with their long range. The method [8] is essentially the generation of two occupancy grids
simultaneously: one based on the maximum sensing range \( r_{\text{max}} \) of the range sensing device and another one based on a more conservative safety distance \( r_{\text{safe}} \). Typical values for \( r_{\text{max}} \) and \( r_{\text{safe}} \) are 20 meters and 3 meters respectively. The result is that the safe region is a subset of the open area. Frontiers can then be extracted on the boundaries of the safe region where the robot can enter the free space. Subsequently, the area beyond the frontier can be estimated directly from the current map by measuring the amount of free space beyond the safe region.

When this information was combined with a movement costs as indicated by a path planner, active exploration is possible. The study [3] demonstrated that the active exploration of the robots can be easily tuned by adjusting the balance between information gain and movement costs. Shifting the balance in favor of information gain has the effect that robots explore mainly the corridors, while shifting the balance towards movement costs has the effect that the robots enter the rooms along the corridors. The algorithm is extensively described in [3].

Fig. 3. Autonomous Multi-Robot Exploration from the north-east corner with two robots. (629 m^2 & 6 victims in 20 minutes)

Another study [4] took into account that this information gained is only valuable when this information can still reach the ComStation, which is not guaranteed with limited communication range that is typical for indoor communication schemes. The communication success rate is difficult to predict, because it depends on the number and nature of obstacles in the line between the robot.
and the ComStation. During exploration the typical signal levels in the environment are learned, to facilitate a more reliable prediction of communication success.

For the 2008 competition the algorithm will be adjusted to include the information available in the a-priori maps. The information gain can be better estimated with the untraversable areas indicated on the a-priori map. The prediction of communication success can be initiated based on the estimate of communication difficulty. The movement costs can be better estimated with level of difficulty of mobility.

5 Victim detection

Camera images can be used to automatically detect victims, independent from the Victim sensor provided by USARsim, as indicated in [5]. This independent information can be used to increase the robustness of the detection. This year the omnicam sensor\(^1\) is introduced in USARsim. An omnidirectional catadioptric camera has some great advantages over conventional cameras, one of them being the fact that visual landmarks (as victims) remain in the field of view much longer than with a conventional camera. This characteristic will be exploited during the competition.

Fig. 4. The UvA robot lab seen through a omnidirectional catadioptric camera (both real and simulated)

The Unreal Engine only allows a limited number of flat mirror. To create an omnidirectional catadioptric camera with a parabolic mirror several security camera are placed at the Effective Viewpoint inside the mirror, and their 2D images are with UV-mapping displayed on the parabolic 3D-surface in front of them. A normal UsarSim camera collects the world as displayed on the parabolic 3D-surface. As can been seen in figure 4, the result is very realistic.

\(^1\) OmniCam package is available on http://student.science.uva.nl/~tschmits/USARSimOmniCam/
6 Validation on Real World Data

To ensure the validity of our scan matching approach with data that suffers from real-world odometric errors and sensor noise, our algorithm is tested on a wide variety of datasets which are available thanks to the initiative of Andrew Howard and Nicholas Roy\(^2\).

The occupancy grid maps illustrated in figure 5 were all created with the standard occupancy rendering techniques from [7]. All presented results are strictly based on scan matching, the SLAM algorithm was purely incremental. The occupancy grid maps can be compared with the original results (see for instance [16, 17]).

The data is collected in indoor environments, with many overlapping feature-rich, dense laser scans. For instance, the 'AP Hill' dataset (figure 5 (a)) was collected for the DARPA/IPTO SDR project when four robots had to explore an unknown building at Fort AP Hill. The dataset is difficult because people were walking around the robots to check their progress. The 'CMU Newell Simon Hall' (figure 5 (b)) is a relatively old and small. The difficulty in this dataset are the straight corridors without many features. The Edmonton Convention Center (figure 5 (c)) is an exploration of quite a large area with lots of open spaces. Stanford Gates (figure 5 (d)) is a huge dataset where the robot traverses several hundreds of meters, both corridors and halls. For the Intel Lab Seattle dataset (figure 5 (e)) the robot traverses several times through the main corridor along which it shortly visits every room bordering on this corridor. One of those loops is mapped. The "MIT CSAIL building" dataset (figure 5 (f)) was used as example by [17] to illustrate how difficult SLAM could be. The "Intel Campus, Oregon" dataset (figure 5 (g)) is collected by a P2DX robot during a tour of the part of the Intel Lab in Hillsboro, Oregon. The last dataset (figure 5 (h)) is collected at our own location, a small tour around a staircase with a Nomad robot equipped with a Hokuyo laser scanner.

These results illustrate the general applicability of our approach and more generally that developments in the Virtual Robot competition can be directly applied to fielded robotic systems.

7 Conclusion

This paper summarizes the improvements in the control software of the UvA Rescue Team after the competition the RoboCup 2007 in Atlanta, where the semi-finals were reached. At the competition in Suzhou will focus on a fully autonomous robot team based on active exploration of the frontiers and automated victim detection the based on color histograms.

\(^2\) The Robotics Data Set Repository (Radish) available on \url{http://radish.sourceforge.net}
Fig. 5. Occupancy maps generated for several datasets (mainly from Radish). The results that are acquired by running a version of our scan matcher, Q-WSM, as an iterative process, without any global optimization at the end of the process. The last figure (h) was recorded at our site with a Hokuyo laserscanner, with a maximum range of 4 meters.
References