Abstract. This year the scan matching routines will be made fully 3 dimensional. A first attempt to incorporate the external library 3DTK to process the 3D point clouds was not successful, because the library did not contain online algorithms. Instead, this year the UvA rescue code will be extended from quadtrees towards octtrees.

Introduction

The RoboCup Rescue competitions provide benchmarks for evaluating robot platforms’ usability in disaster mitigation. Research groups should demonstrate their ability to deploy a team of robots that explore a devastated area and locate victims. The Virtual Robots competition, part of the Rescue Simulation League [1], is a platform to experiment with multi-robot algorithms for robot systems with advanced sensory and mobility capabilities.

To be able to efficiently coordinate a team of robots in a disaster situation many state-of-the-art robotic techniques have to be integrated. Our approach is extensively described in previous Team Description Papers and aggregated in a Technical Report [2]. The later report gives an overview of the publications for the period 2008-2012. In addition, our team had several more recent publications [3–12]. In this paper we will concentrate on this year’s innovations.

1 Team Members

USARCommander was originally developed by Bayu Slamet and all other contributions have been integrated into this framework. Many other team members have contributed to perception and control algorithms inside this framework.

The following contributions have been made and will be made this year:

- **Mustafa Karaalioğlu**: 3D scan matching
- **Arnoud Visser**: Planar 3D mapping
2 Planar 3D mapping

One of the main challenges faced in the competition (and for robots in general) concerns building a map of the environment as the robot explores it. The Amsterdam-Oxford team currently has software that enables a virtual robot to build a 2-dimensional map from sensor data. Although useful in a number of situations, this is potentially quite limiting in the real world, as search and rescue operations are unlikely to take place in perfectly flat environments. Therefore, the aim of this project is to extend the mapping capability to 3-dimensional space. This would give rescue workers a better idea of the layout of the environment, and would help to highlight features and hazards that would not be apparent in a 2D map.

In a previous attempt by Nelson [13] to build a 3-dimensional map inside USARSim, the classical ICP algorithm [14] was implemented efficiently with a Kd-Tree, an approach already advocated by Rusinkiewicz and Levoy [15]. For efficiency reasons, not the full 3D point-cloud was used by Nelson. Instead, only the edge-points of surfaces were used, which gave an efficiency improvement of a factor 100. The ICP algorithm tries to reduce the distance between the 3D-datapoints by finding the optimal translation and rotation vector, which implies a full 6D localization.

Yet, in the nomenclature of Borrmann et al [16] this is planar 3D mapping and not full 6D SLAM, because the 3D point-cloud consists of slices of a rotating 2D laser scanner. When acquiring this data while moving, the quality of the resulting map crucially depends on the pose estimate that is given by inertial sensors. In principle, the probabilistic methods from planar 2D mapping are extendable to full 3D mapping with 6D pose estimates. Yet, for 3D point-clouds it is essential to have a good strategy for reducing the computational costs of matching.

In the approach of Nelson [13], the 3D map is built based on pairwise ICP, which gives good results for local maps, but in the end registration errors sum
up. Borrmann et al [16] solve this by adding a loop-detector in the code, which indicates when a place is visited for a second time. At that moment a 6D graph optimization algorithm for global relaxation based on the method of Lu and Milios [17] is employed.

The benefit of using the 3D Toolkit is not only that it contains Lu and Milios’ SLAM algorithm, but in addition that the whole implementation is highly efficient, for instance the data structure for the nodes is 8 times smaller than the data structure used in PCL [18].

![Image](image.png)

**Fig. 2.** Example of a 3D scan of a passage in Dagstuhl castle, recorded during the workshop ‘Towards Affordance-Based Robot Control’, 2006. Courtesy Andreas Nüchter [19].

Currently, an attempt is made to extend the implementation of the Weighted Scan Matcher based on quadtrees [20] to an implementation based on octtrees using a similar data structure.

### 3 Optimalizations

This year the robustness and responsiveness of our user interface has been greatly improved by a number of bug fixes in the code. During the competition at the Iran Open 2014 finally a memory leak was found which prevented us from scaling up the control of larger robot teams. After the Iran Open competition the responsiveness was improved by using event notifications instead of CPU-consuming event polling. In addition, the camera subview is now cut out of the Unreal Tournament window at the robot side (instead of the user interface side). This is both an improvement in realism (the robot camera does not have the whole Unreal Tournament overview) and in efficiency.
4 Conclusion

This paper summarizes the plans for improvement of the algorithms of the UvA Rescue team. In 2013 our team was active in the Darpa Robotics Challenge. Many developments inside our framework are not only valuable inside the Rescue Simulation League, but could also be valuable for the Soccer Simulation, the RoboCup@Home, RoboCup@Work [21] and the Standard Platform League [22]. For the Virtual Robot competition, developments towards full 3D mapping are important to bring the competition closer to field studies.

References