This is a preprint of a paper which appeared in the Proceedings of the 10th IEEE International Symposium on Safety Security and Rescue Robotics (SSRR 2012), Nov 5-8, 2012, College Station, Texas

Application Independent Supervised Autonomy

Karen Petersen and Oskar von Stryk
Technische Universität Darmstadt
Hochschulstr. 10, 64289 Darmstadt, Germany
{petersen, stryk}@sim.tu-darmstadt.de

Abstract — In this paper, a general methodology is presented, that allows a human to supervise autonomous robots for different types of tasks. The commands to advise the robots are classified in two general categories: commands that modify the tasks, and commands that modify the allocation of tasks to robots. The method is not tailored to a specific task allocation algorithm, because only the input data for the algorithm are modified. In that way, the task allocation is influenced implicitly, without the need to change the actual algorithm. This implies that different approaches to task allocation can be exchanged transparently, which enables to apply the supervision concept to fundamentally different problem classes. Experiments in simulation with a rescue robot show, that the robot's performance with respect to the number of detected victims and the covered area can be significantly improved.

Keywords: *supervised autonomy, task allocation*

I. Introduction

In many robot applications it is impossible to blindly trust autonomous robots. Existing solutions for supervision of robots often disregard the specific capabilities of robots and humans (e. g., fixed task assignments [3]), involve phases of teleoperation, where the human cannot attend other robots (e. g., [4]), or support only a limited amount of scenario-specific operations (e. g., predefined autonomy modes [5], assignment of waypoints [2]).

To overcome this, we define a set of application invariant commands and present a method how these commands can be integrated into the robot control software by modifying the input data for the task allocation algorithm.

Different approaches to task allocation, like centralized methods, market based methods, or behavioral approaches can be appropriate, depending on the available infrastructure [1]. Hence, by modifying only the input data instead of the whole algorithm, the method can be combined with different task allocation methods and therefore stays independent of the application.

Potential application areas are surveillance and monitoring, factory automation, or robot soccer. In this paper, the method is applied to urban search and rescue (USAR).

II. Supervision Concept

A. Application Independent Supervisor Commands

Although it is not always mandatory for mission achievement, input from a supervisor can often facilitate or speed up the accomplishment of a task.

The commands used in this context are:

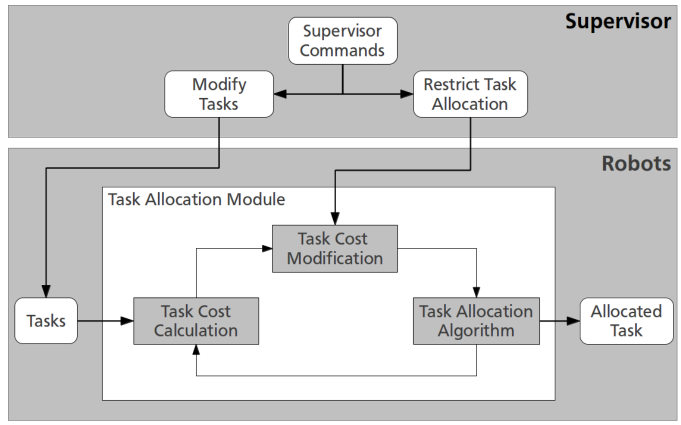

Fig. 1. The supervisor commands modify the tasks and the task costs. Only the input data are changed, the task allocation algorithm itself is not modified.

- Add and delete tasks
- Adjust task parameters
- Manually assign tasks to robots
- Release robots from single tasks or task types
- Switch between autonomy and teleoperation
- Define preferred task types for a robot

These generic commands are not tied to a particular application. We subdivide them into two categories: changes to the tasks and restrictions on the allocation of tasks to robots.

B. Modification of Tasks and Mission Details

The first commands can be applied by modifying the set of tasks and synchronizing with all team members. This functionality is already provided, because the robots themselves can add, change, and delete tasks. However, it has to be considered that changes from the supervisor must not be overwritten autonomously afterwards.

Apparently, there is no need to change anything inside the task allocation module. Instead, only the input data for this module are modified by the supervisor (Figure 1, left).

C. Restrictions on Task Allocation

The last four commands restrict the allocation of tasks to robots. Instead of adapting a task allocation algorithm, we propose to rather modify the costs for executing the tasks, to cause any cost-based task allocation algorithm to comply with the commands. Hence, different algorithms for task allocation can be exchanged transparently.

To modify the task costs according to the supervisor's commands, a module that modifies the calculated costs is inserted in between the cost calculation and the task allocation (Figure 1). For example, tasks that must not be assigned to a robot have infinite costs, while tasks that are assigned to a robot have zero costs. In that way, only the input data for the task allocation are changed, and the supervisor commands are implicitly addressed. The task allocation algorithm itself remains unchanged.

III. Applications and Results

The aims is to enable a human supervisor to support autonomous robots. Therefore, we compare the results of a purely autonomous robot to those of a supervised robot. The autonomous baseline is the solution by Team Hector Darmstadt (`www.gkmm.tu-darmstadt.de/rescue`), which won the "best-in-class autonomy award" and placed second in the overall rescue robot competition at RoboCup 2012 .

A. User Interface

The robot is running ROS (`www.ros.org`). For the experiments, we used rviz (`www.ros.org/wiki/rviz`) and plugins for rqt (`www.ros.org/wiki/rqt`). In rviz, the map is enhanced with overlays of the current position of the robot, the robot's trajectory and interactive markers for open tasks. The rqt plugins provide buttons for interactions and a list of tasks, that highlights the currently allocated task.

Adding, deleting, and modifying tasks is done by interactive markers in rviz using menus and by dragging the markers. Assigning and forbidding tasks can be achieved either via the menu of the markers, or via the tasks list in the rqt plugin.

B. Experiment Setup

The experiments were conducted in simulation using gazebo (`http://gazebosim.org/`). Within gazebo, we modeled the RoboCup German Open rescue arenas from 2011 and 2012. For each of the two arenas, 5 victims were placed randomly (but fixed) at typical victim locations from the competitions. In both arenas, each of the 5 participants supervised one mission, and 5 missions were conducted fully autonomous.

C. Results

The performance metrics for the experiments are the number of victims found and the area covered by the robot. Because the laser range finder (LRF) has a much longer range than the thermal camera, besides the coverage of the map also the coverage of a 1.5 meter radius around the robot's path is considered.

The results are summarized in Figure 2. An unpaired t-test showed that the number of detected victims and the coverage with respect to the robot's path improved significantly. This indicates, that the increased number of detected victims with supervisor support is potentially due to the increased coverage of the robot's path in the arena.

It strikes out, that the autonomous robot discarded 4 victims because the calculated position to approach the victim was either not reachable, or the victim was not visible for the robot from this position. Similar situations also occurred several times in the supervised missions, however, the participants were able to support the robot in finding the victim by moving the search position to a more appropriate location.

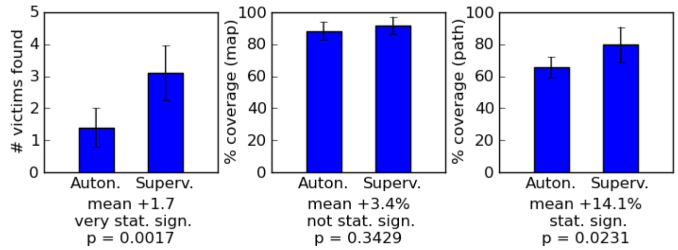

Fig. 2. Comparison between the autonomous and the supervised missions.

IV. Conclusion and Outlook

The methodology presented in this paper allows a human supervisor to remotely support autonomous robots in achieving their tasks. Unlike other methods, it can be applied to a variety of different scenarios. The exemplary application to USAR is demonstrated in a simulated environment.

The applied set of supervisor commands is independent of the application. Furthermore, the method is independent of the applied approach to task allocation, because only the input data are modified. This allows to transparently exchange the task allocation algorithm, to always select the algorithm that fits best to the current needs of the application.

Experiments with a simulated USAR robot substantiate the hypothesis, that support from a human supervisor can significantly improve the performance of autonomous robots.

In the authors' group, the presented method and the same commands are also used for autonomous soccer robots. This shows, that the method is not application specific, and can also be applied to robot teams. Future work will include user studies with robot teams in both, robot soccer and USAR.

Acknowledgment

This research has been supported by the German Research Foundation (DFG) within GRK 1362 "Cooperative, adaptive and responsive monitoring in mixed mode environments".

References

[1] M. B. Dias and A. Stentz. A comparative study between centralized, market-based, and behavioral multirobot coordination approaches. In *Proc. of the IEEE/RSJ int'l conf. on Intelligent Robots and Systems (IROS '03)*, pages 2279–2284, 2003.

[2] Y. Nevatia, T. Stoyanov, R. Rathnam, M. Pfingsthorn, S. Markov, R. Ambrus, and A. Birk. Augmented autonomy: Improving human-robot team performance in urban search and rescue. In *Proc. of the IEEE/RSJ int'l conf. on Intelligent Robots and Systems (IROS)*, pages 2103–2108, 2008.

[3] P. Scerri, P. Velagapudi, K. Sycara, H. Wang, S.-Y. J. Chien, and M. Lewis. Towards an understanding of the impact of autonomous path planning on victim search in usar. In *Proc. of the IEEE/RSJ int'l conf. on Intelligent Robots and Systems (IROS)*, pages 383–388, 2010.

[4] B. Sellner, F. Heger, L. Hiatt, R. Simmons, and S. Singh. Coordinated multi-agent teams and sliding autonomy for large-scale assembly. *Proc. of the IEEE special issue on Multi-Robot Systems*, 94(7):1425–1444, 2006.

[5] J. Wang and M. Lewis. Human control for cooperating robot teams. In *Proc. of the ACM/IEEE international conference on Human-robot interaction (HRI)*, pages 9–16, 2007.