Autonomous Distributed Rescue Robot System

Jon Rönnols, Pär Johansson, Jakob Carlström
Department of Information Technology
Box 337, 751 05 Uppsala, Sweden
Uppsala University
Email jakob.carlstrom@it.uu.se

Abstract

This paper describes the autonomous distributed rescue robot system developed by the Hermod Project at Uppsala University, Sweden. A hierarchical model for robot team behaviour is used to achieve efficient negotiation of disaster areas.

Summary

The Hermod project has developed hardware and software for robots targeting the RoboCup Real Rescue League. Ultrasonic and IR sensors are used for obstacle detection and positioning, pyro-electric sensors on rotating arm supply victim detection and positioning, and dead reckoning is the chief method for robot positioning.

An arbitrary number of robots cooperate to solve a rescue task. Robot behaviour is controlled by a software hierarchy, shown in Figure 1. Each robot is initialized with state variables including a unique priority. Among the robots which manage to establish contact on a wireless network, the one with highest priority is elected master. In case of communication failure with the master robot, the robot with second highest priority will become the new master, etcetera.

The master partitions the rescue task; i.e., the unknown area, into sub-tasks consisting of smaller exploration areas, and distributes them evenly to all robots.

Master Selection	(distributed)
Task partitioning	(master)
Sub-task and map distribution	(master/slave)
Sub-task planning	(distributed)
Sub-task execution	(distributed)
Emergency action	(distributed)

Figure 1 Robot software hierarchy

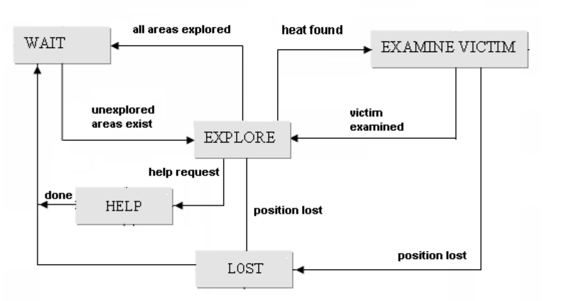

Figure 2 Robot Finite State Machine

On receipt of its sub-tasks, each robot selects its closest exploration area and uses a modified A*-algorithm [1] to plan the shortest path here. The robot follows the planned route, using its ultrasonic, IR and pyro-elecric sensors to create a local map and a local heat map. These are sent via WLAN to the master robot that synchronizes them into a global map and a global heat map, which are broadcast to all robots.

As shown in Figure 2, the robot is controlled by a Finite State Machine (FSM) during task execution. Negotiation of exploration areas uses greedy search of regions with unknown state. Emergency actions, aiming at avoiding obstacles and holes, may temporarily override the behaviour ordered by the FSM. On completion of a sub-task, the robot selects the nearest remaining sub-task, until all sub-tasks have been completed.

To improve the efficiency of the development, a simulator for the robot environment and its sensors was developed, enabling parallel development of hardware and software.

References

[1] Stout, B. The Basics of A* for Path Planning, *Game Programming Gems*, pp. 254-263, Charles River Media, Hingham, USA, 2000.