Remote-operated robot swarm for measuring an environment

Antti Tikanmäki, Janne Haverinen, Anssi Kemppainen, Juha Röning

Robotics Group, Intelligent Systems Group,
University of Oulu, Finland
antti.tikanmaki@ee.oulu.fi
+35885532537, P.O. BOX 4500
90014 University of Oulu, Finland

ABSTRACT

In this paper, we describe a mobile distributed multirobot system and how it is used to measure an environment semiautonomously. The system contains a swarm of robots and a remote user computer for a human operator. The swarm has two kinds of robots; measurement robots and a group coordinator that delegates tasks to each robot. The measurement robots have different kinds of capabilities and sensors. The set of sensors can be changed, and therefore different kinds of measurements can be done remotely. The system is semi autonomous and a human acts as a supervisor. The human operator controls the group's operation by sending higher-level commands and tasks to the group coordinator. The coordinator delegates a task to each robot in the swarm, providing effective operation in the swarm and each individual robot. Each robot does its own task autonomously. To be able to use a system with different kinds of robots with different kinds of capabilities, a dynamic interface is needed. The usage of the Property Service Architecture provides this possibility. The same architecture has already been used in several other systems. This measurement swarm can be a part of a larger system and can interoperate with other services in a distributed system.

Keywords: Remote operation, robot swarm, miniature robot, Property Service, networked robots

1. INTRODUCTION

One of main applications of mobile multirobot systems is to act as different kinds of tools to ease human work. One of these tasks is routine measurements of larger areas that have been traditionally done by a human using a hand-held device. For example, humidity measurements, temperature mapping of a room, and lightness measurements in different parts of the room, are time-consuming tasks in larger areas. The measurements are usually done by hand using a single measuring device or statically placed sensing nodes that form a sensor network. Sensor networks are useful in cases where measuring times are long and placement of the sensors is permanent, and if the measured space is rather small. However, when the user wants to measure the humidity of a room only once, it is more reasonable to use a moveable unit. When the space is large and the user wants dense and accurate measurements, using a mobile robot swarm becomes a very useful tool. Measurements in places that are not reachable by a human, like space or hazardous disaster areas, are also applications where mobile robots can make the measurements.

In applications where continuous and more permanent measurements are needed, it is more reasonable to use static measurement nodes as sensor networks. Several different kinds of wireless sensor networks are already commercially available, like Bluetooth. One of the most promising techniques is the latest development of ZigBee [1] embedded networks. In comparison with sensor networks, simple autonomous mobile robots are used as measuring nodes in more dynamic applications where it is not reasonable to use statically placed nodes. Mobile robots have been developed also for this purpose in several research projects like [2] and [3], as have several methods for solving problems of coordination [4], task allocation [5], and area coverage [6]. The major differences of our system in comparison with these systems are its high adaptivity to different kinds of robots, development of small modular robots “Minirobots” and distributed software architecture development called Property Service Architecture.

This paper describes a mobile multirobot system capable of collecting measurements from user-defined unknown areas. The system has been developed in two main steps. In the first stage, a robot swarm simulation, RoboFarm [7], was used as a test bed for system component development. In the second stage, the full functional system was used with a group of real robots measuring a real environment.

The major development guideline has been to create system that is operable with different kinds of robots. This is done by using a Property Service Architecture [7]. The Property Service Architecture for robot swarms provides the possibility to create a swarm of robots using different kinds of robots, with wheels or legs and different kinds of sensors. As the capabilities of the robot increases, new properties are added to the service. The swarm system takes advantage of these features in task delegation.

This paper is structured as follows: In section 2, each system component is introduced on a general level. Section 3 gives detailed information on the robots used, communication, and the property service implemented in the experiments of this work. Section 4 gives the results of the experiments.

2. SYSTEM COMPONENTS

2.1 Overview

In this work the robot communication network has two kinds of nodes: measuring robots and coordinators. The swarm has one or several group coordinators. The coordinator can be software running on the user’s computer or one of the robots in the swarm can specialize to act as a coordinator. Figure 1 shows an overview of the system components. "R" is a measuring robot, "C" is a coordinator and "User" is a human operator.

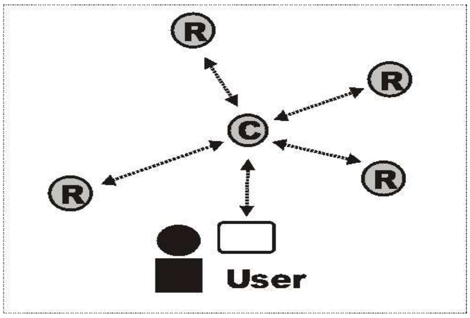

Figure 1, Simplified overview of the system

Measuring robots may have different kinds of capabilities related to sensing and moving. The coordinator may be one similar robot with a special task or it can be a static communicator without moving capability. The coordinator may also be a software module running on the user’s computer. The user has a graphical interface where tasks are given to the swarm and measurements are visualized.

Data delivery and communication can be done in several ways. Either each robot sends its measurements to the coordinator which forwards the combined group status to the PC, or each measurement robot sends the measurement data to the PC. Both ways are possible in our system. In our system, the coordinator is used as a location measure that measures the relative locations of each measurement robot. The sensor output of the coordinator is therefore the locations of the measurement robots.

Measurements from each robot are sent and stored on a special service called *spatial measurement data storage*. The storage gathers the information and stores it in common data storage.

2.2. Measurement robots (R)

Each measurement robot has a set of sensors and several primitive behaviors that can operate the actuators of the robot. It can navigate around the environment, avoid obstacles and execute given paths. The set of sensors can be changed without the need to modify the system. This sets some dynamic requirements on the interface of the robot, which are solved with the Property Service Architecture. This is explained later.

The primitive behaviors of the robots are: random drive, wall following, seek target, move to target or position. These behaviors are multipurpose as, for example, the target can be changed. The behaviors can also be combined into more capable behaviors. As the capabilities of the robots increase, the same system can be used for various other tasks where robots are needed to manipulate the environment.

2.3. Coordinator (C)

The coordinator has several tasks. It delegates roles to each measuring robot, acts as a link between the user and each robot, and acts as a reference coordinate (origin) for location measurements.

2.3.1 Measurement procedure

A typical measurement proceeds as follows: The coordinators first find a good location. This can be, for example, the center of the room. Each member of the group (measurement robots) follows the movements of the coordinator in a defined formation. The human operator needs only to order the coordinator and other the robots will follow it.

As the coordinator reaches the location, it stops and starts to coordinate the roles of the other robots in the swarm. In the case of several coordinators, each robot is assigned to one coordinator. The coordinator receives all the measured information and the location of measurement robot relative to the coordinator's location. An accuracy of position estimate of each measurement robot is also sent to the user software.

In the simplest case, the coordinator sends direct destination points to each robot. As the coordinator has a control for each robot, it provides a possibility for maximum coordinated coverage of the monitored space. In case the user has a certain area of special interest, the coordinator may command denser coverage of that area by sending more robots to that position. Similarly, the coordinator can generate "moving target" positions for group of robots. This provides sort of measurement formation on movement of the group as each robot follows its own target.

2.3.2. Task delegation

The coordinator assigns roles to each robot depending on the higher-level task of the swarm. The number of robots for each role is set by the human operator, or by using a certain optimization method. For example, if the swarm's task is to "seek and gather", 30% of the robots are assigned to drive around and find targets and 70% of the robots are assigned to move targets to a defined location. In a heterogeneous swarm, a robot's capability defines what role can be assigned to it. For example, faster robots become scouts and stronger robots become transporters. Also, for example, a robot's battery power level is used in deciding the of robot's role. In practice, this is done by requesting the properties of each robot to determine its capability.

2.4. Measurement Data Storage

Measurements of the environment are sent and stored in the coordinators memory or in a special service called *spatial measurement data storage*. Each measurement robot delivers a data packet that contains measurements and possibly location information.

The data storage operates as follows. Each received measurement is put in a vector buffer as it arrives. The storage continuously processes received data so that it can optimize its content. Externally the data can be seen as a grid, but internally no fixed grid is used. The storage combines measurements from the same area to one measurement unit according the defined measurement cell size. For example, if the cell dimensions are 10cm x 10cm, all the measurements from the same cell are combined into one measurement unit in the storage. The main benefit in comparison with a fixed grid (like an occupancy grid) is that only the areas that are covered are stored in the storage and area range is not limited as the robots travel farther away. In the fixed grid case, the grid must be resized and it quickly becomes huge, event though it contains only a small amount of data. Figure 2 illustrates the principle of each storage type and their difference.

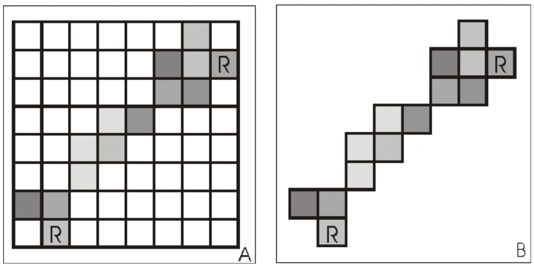

Figure 2, a compare of occupancy grid and expanding data storage

The content of data storage may be for example:

(1.0 2.2) 12.5
(1.1 2.2) 12.2
(1.2 2.2) 11.2
(1.3 2.3) 10.2,

where the first two numbers in each row are location coordinates and last number is a measured parameter (for example, temperature).

The main drawback of using spatially expanding data storage is that each measurement unit requires more memory as the coordinates of the measurement are included in the data unit. In the fixed grid case, the coordinates are defined by the grid parameters as the grid is usually implemented as an array. However, the advantage of memory saving in a larger grid is essential.

Spatially expanding data storage also contains several built-in functionalities. It *propagates* values, converting measurements into a fixed size grid. The conversion is done when a certain area has been fully measured. This is done to save memory. The measurements can be converted into a grid also when the area is almost fully covered, by filling in a missing measurement by using the propagation method. Furthermore, internally the data storage may contain several grids, improving its performance in searching for values for a requested coordinate and reducing its memory requirements. In propagation, the storage can spread the measurement values to a neighbouring unit so that the unit gets values even though they are not actually measured. Of course this feature can be selected by the user in the case it is needed.

Spatial measurement storage can be a part of the coordinators software, it can be located in the user interface program, or it can be a stand-alone service in the Property Service Architecture. In this work, the last one is used.

2.5. Property Service Architecture

An essential part of the system is the software architecture for creating interfaces for each resource in the system. The Property Service architecture [7] has been used. In the property service interface, each property (feature or functionality) is used with SET and GET methods. All the properties currently available in the service are requested using a "GET properties" command, returning a list of property names. This may vary over time as new features can be available and the service may have expanded functionalities or capabilities.

Each measurement robot has its own property service accessed through a radio link. The robot's sensors and actuators are accessed using properties, and well as different behaviors are also activated and configured using properties. As the properties may be changed dynamically during operation, the capabilities of the robot can be changed. The simplest mobile robot service must provide at least the properties listed in Table 1.

Name	Purpose
properties	list of current properties
name	ID of the robot
position	current position of the robot
movement	current movement
behaviors	list of behaviors
sensors	list of sensors on the robot
targets	for robots behaviors

Table 1. Basic properties for a mobile robot service

Each of these properties may contain several subproperties. For example, the property "*behaviors.moveto*" is a property for activating a robot's navigation. By changing the targets for moveto behavior, different kinds of behaviors can be achieved. A robot may have a property called "*sensors.temperature*", which returns a measured temperature value.

When new sensors are added to a robot or a new algorithm is uploaded to a robot, each of these shows up as a new property in the service. As the capabilities of the robot increases, for example, to manipulating an environment, the set of properties is expanded. As the coordinator checks available properties, it can delegate a more demanding task to the robot.

Similarly, each main component in the system, like the user interface, and data storage, can be used with the property service. The swarm also has its own property service. In the Property Service Architecture, several services can be grouped into one service and they can be used through that service. All the properties of each subservice become subproperties of this grouped service. In this case, grouped service is called a Swarm Property Service. In addition to subproperties, swarm service has also own properties related to groups operation. For example, the group services "*targets*" property is a target of the groups operation and the behaviors of the group. Internally the swarm service delegates the task to each sub service.

2.6. User Interface

The user interface is used for two main purposes; to visualize the current state of the system, by visualizing the measurements and operation of the robots, and to act as a control interface for the swarm of robots.

The user interface has "game-like" functionalities for selecting a group of robots and ordering them to do tasks. The user can mark areas of interest and order a group of robots to go and do the measurements. Using Right mouse button, the user can select a behavior for selected robots from popup-menu. Figure 3 shows an example of GUI. A human has marked an area of interest with a circle. Several robots, visualized as smaller circles, are available for measuring. As the measuring continues, more data becomes available in the view. The measurements can be sorted by each individual robot and by different layers; the relation to the measured feature can be visualized. The user can also save measurement layers as an image.

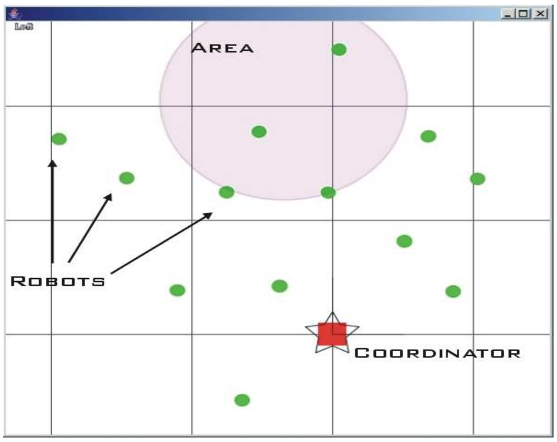

Figure 3, Example of a GUI for controlling multiple robots.

Figure 4 shows an example of a GUI in operation. Measurements collected from the robots are shown as circles on the grid and the shade of color expresses the value of the measured data. The coordinator is located at the origin of the image.

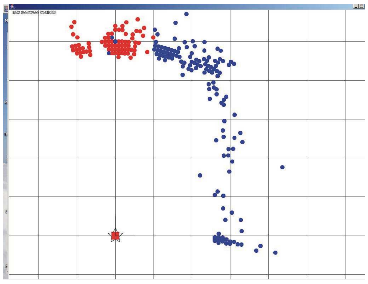

Figure 4, Visualization of robot locations in a user interface

3 MEASUREMENT SYSTEM DESCRIPTION

3.1. Overview

The system was created and tested first with a simulated environment (RoboFarm) and the same software was later used with real robots. The system was developed so that it is possible to use either simulated or real robots without having to change the system. An overview of software components is shown in Figure 5. The interface for a swarm and for a simulated swarm is similar, a *swarm property service* interface.

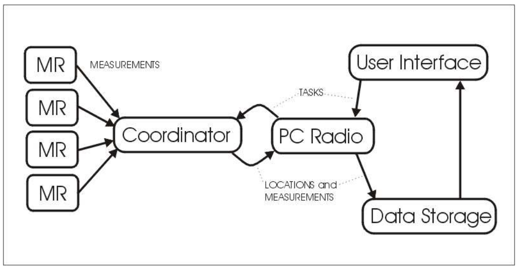

Figure 5, Software components

3.2. Simulated swarm

In the first stage, a multirobot simulator was used to develop the system components. This RoboFarm [7] provides a possibility to simulate large swarms of robots and their operation a defined virtual world. The world can contain objects and different world parameters, like temperatures in different areas, can be defined using bitmaps.

In the simulator, the environment was measured with up to 200 simulated robots. The swarm was divided into 10 groups with one coordinator in each. The human operator controlled each coordinators target locations and assigned task to the subgroups. Each coordinator then delegated subtasks to each measurement robot in the group. The communication was simulated in a simple way by defining the maximum hearing distance and normally distributed random location error for local location estimations.

The simulation environment was defined with a large image, where each pixel corresponded to a measured value on environment. As each robot did it’s measuring, the value of the pixel on that coordinate was returned as the sensor’s value.

3.3. Real robot experiment

3.3.1 Minirobots

In the real robot experiments of this work, Minirobot's [8] were used. The robot is 80mm in diameter and has two wheels and a differential driving and steering mechanism. The Minirobot has modular electronic architecture. Each module provides one or several functionalities, and they are stacked together on the robot. Each module has its own microcontroller, and they communicate through a serial bus with each others. Figure 6 shows the robot with its modules stacked together.

The current sensor set of the robots was designed to supporting swarm robotics research. Unique stereo camera system was of particular importance, as it can be used to estimate distance, record stereo image sequences, recognize objects and colors, produce stereo panoramic views by combining the observations of different robots, and for active sensing, for example. It is also important to have an on-board camera system to provide maximum flexibility and to enable the implementation of a large-scale multiagent system.

Figure 6, The mini robot with color vision

In addition to the stereo camera system, the robot has two other sensor modules: an environment module, and the infrared sensor module. The environment module has sensors for measuring various environmental properties. The sensor set of the module consists of a thermometer, a 2-D accelerometer, a visible light detector, a digital compass, and a real-time clock. The visible light detector, the digital compass, and the thermometer might be used for particle filter-based localization [9], for example. The accelerometer detects vibrations and forces affecting the robot. The real-time clock can be used to power up the robot at regular intervals from a low-power sleep mode, and to time stamp observations such as pictures taken of a moving target.

The infrared sensor module consists of nine infrared emitters and detectors. It is used as a proximity sensor and to implement an infrared based communication link. In addition to the infrared communication link, the robot has a radio transceiver module enabling inter-robot radio communication and wireless software uploading, which is an important feature in a large-scale multirobot system.

3.3.2 Robot communication

Each robot has a radio module that can be used to communicate with each others. The radio module is based on an nRF905 multiband transceiver designed for 433/868/915MHz, and it implements the inter-robot radio communication link. The radio module has 512kB of on-board Flash memory for temporarily storing uploaded application code.

Each robot has a unique 8-bit agent address and a physical radio address. The sensor data of a robot can be read thought the radio link to a host computer for storing and further analysis. The host side radio link can access any robot by defining the agent and the radio addresses of the robot. The radio link can also be used for wireless software uploading, which is an important feature in a large-scale multi-robot system. With this feature, it is possible to change the behavior of the robot by uploading new application software for the modules on-the-fly. This enables the replacement of the vision algorithm running in the camera module, for example.

In the experiments, each robot has its own time window for sending sensor data. On its own sending turn, each robot sends a vector of values. The vector contains all the measured values in a defined order in one data packet. Each robot has its own window in the communication channel and the windows are defined by the coordinator. When it is the coordinators sending turn, it sends the current locations of each measuring robot to PC radio. All the measurements and location information are combined in data storage and each type of data is stored on own data storage layer. The human operator can then select the desired layer in the user interface to view different collected data maps.

3.3.3. Location system

Relative locations among a group of miniature robots are obtained using an infrared location system [10]. In the location system, the robots use intensity and bearing measurements of received infrared signals to estimate the positions of other robots in polar coordinates. In addition, each robot has a particular modulation frequency from which they are recognized. The location system implemented enables relative position estimation among a group of robots without exchanging position estimates. This is advantageous, since the robots are able to maintain formation also in the absence of a radio link. Figure 7 shows the location module for a Minirobot.

The location system performs position estimation by rotating a beam collector at a constant rotation speed and by measuring the bearing and intensity of the received signal. Infrared signals are received through a small aperture in the beam collector enabling accurate bearing measurements. In order to maximize the measurement range, infrared radiation is reflected sideways into a uniform zone using a conical mirror. In the latest type of infrared location system, lenses are used to amplify the intensity of the received signal. This improves noise tolerance and increases the measurement range.

Figure 7. Infrared location module for a Minirobot.

This is used in the experiments to measure relative positions between the coordinator and the measuring robots. In the latest type of location system, the system performs robot recognition faultlessly within the measurement range of up to 5 metres. The standard deviation of angular estimates remains close to 1 degrees for all measurement distances and angles. By contrast, the standard deviation of radial estimates depends highly on the measurement distance, since the intensity of the received signal is inversely exponential to the measurement distance. For this reason, small variations in the received intensity and small noises will significantly increase the standard deviation when the located robot is far away.

3.3.4. Test environment

The same measurement system was used with real robots, using a group of six Minirobots. The swarm was successfully used for distributed measurement of lightness and temperature in the 5 m x 5 m area in the research room. The swarm had one coordinator that delivered all the measurements to PC software that was used as an operator tool and a visualization tool for group measurements. The measurements were then visualized as an image with shades of color representing the level of temperature and lightness in the space.

4. CONCLUSIONS

The developed system provides an easily expandable measuring system that can be used in even larger areas. Simulations showed that the advantage of increasing the number of the robots is that the time spends for measuring the whole area can be decreased, as more than one robot is used. Based on the simulation tests, one human operator is capable of operating a large swarm of robots, as they are semiautonomous and the coordinators do the dirty work.

Instead of using static sensor networks, these mobile measurement platforms can be used to make more dynamic measurements. The measurement area can be changed more easily during operation, and the swarm can cover the area much faster than a single measuring device. Robots can be ordered to return to "home base" after the measurement has been made. In non-mobile sensor networks, the operator must collect all the sensors after the measurement. What is more, the same system can be expanded to also manipulate the environment.

The Property Service Architecture provides a simple interface for control different kinds of devices and robots. Minirobots with rather limited capabilities were used in the experiments. Futher, the same system (with GUI and control architecture) can be used for group of more advanced robots as the Property Service interface and set of properties for them are the same. For example, outdoor robots with a GPS location system could be used to achieve similar operation. As the system has been developed modularly and using the Property Service Architecture, the higher-level components of this system can be used in different kinds of systems.

5. ACKNOWLEGEMENT

This work has been partly funded by the Academy of Finland.

6. REFERENCES

[1] Zigbee specification, ZigBee alliance, http://www.zigbee.org

[2] Sahin E., Franks N.R. “Measurement of Space: From Ants to Robots” In *Proceedings of WGW 2002: EPSRC/BBSRC International Workshop Biologically-Inspired Robotics: The Legacy of W. Grey Walter*, pages 241-247, Bristol, UK, August 14-16, 2002.

[3] K. Konolige, C. Ortiz, R. Vincent,A. Agno, M. Eriksen, B. Limketkai,M. Lewis, L. Briesemeister, E. Ruspini, ”CENTIBOTS- *Large Scale Robot Teams”, Artificial Intelligence Center SRI International Menlo Park, CA, 2003*

[4] Baldassarre G., Parisi D., Nolfi S.In S. Schaal, A. Ijspeert, A. Billard, S. Vijayakumar, J. Hallam and J-A. Meyer Eds.), “Coordination and Behaviour Integration in Cooperating Simulated Robots” *From Animals to Animats 8: Proceedings of the VIII International Conference on Simulation of Adaptive Behavior*, pp. 385-394. MIT Press, Cambridge, MA 2004

[5] Labella T.H., Dorigo M., Deneubourg J.-L. In A.J. Ijspeert, D. Mange, M. Murata, and S. Nishio, Efficiency and Task Allocation in Prey Retrieval *Proceedings of the First International Workshop on Biologically Inspired Approaches to Advanced Information Technology (Bio-ADIT2004)*, Lecture Notes in Computer Science, pages 32-47. Springer Verlag, Heidelberg, Germany. 2004

[6] Andrew Howard, Maja J. Mataric´, and Gaurav S. Sukhatme, "Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem," In *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems*, pp. 299--308, 2002.

[7] Tikanmäki A, Röning J (2004), Advanced Remote Operation of swarms of Robots, 2004 SPIE / Intelligent Robots and Computer Vision XXII, Philadelphia, USA

[8] Haverinen J, Parpala M & Röning J (2005), A Miniature Mobile Robot With a Color Stereo Camera System for Swarm Robotics Research, Proc. IEEE International Conference on Robotics and Automation (ICRA2005), Apr 18-22, Barcelona, Spain, p. 2494-2497.

[9] S. Arulampalam and S. Maskell and N. Gordon and T. Clapp (2002), A Tutorial on Particle Filters for On-line, Non-linear/Non-Gaussian Bayesian Tracking, , Feb 2002, IEEE Transactions on Signal Processing, volume 50, number 2, p. 174-188

[10] Kemppainen A. (2005) An Infrared Location System for Mobile Robots, Master Thesis University of Oulu, Department of Electrical and Information Engineering