MULTI-TOUCH INTERFACE FOR CONTROLLING MULTIPLE MOBILE ROBOTS

複数台の移動型ロボットを操作するためのマルチタッチインタフェース

by

Jun Kato

加藤 淳

A Senior Thesis

卒業論文

Submitted to

the Department of Information Science

the Faculty of Science, the University of Tokyo

on February 10, 2009

in Partial Fulfillment of the Requirements

for the Degree of Bachelor of Science

Thesis Supervisor: Takeo Igarashi　五十嵐 健夫

Associate Professor of Information Science

ABSTRACT

We must give some form of an order to robots in order to have the robots do a complex task. An initial instruction is required even if they do their tasks autonomously. We therefore need interfaces for the operation and teaching of robots. Natural languages, joysticks, and other pointing devices are generally used for this purpose. These interfaces, however, have difficulty in operating multiple robots simultaneously.

We developed a multi-touch interface for controlling multiple mobile robots to solve this inconvenience. Our interface uses a top-down view from a ceiling camera and has two operating modes. One way is named "Direct Operation" in which users' fingers specify the motion of individual robots and the other is "Vector Field Operation" in which they modify a global vector field followed by robots.

This paper describes the user interface and its implementation, the results of a user study. We compared two modes of the multi-touch interface. The results show that each mode has different characteristics and they make up for each other's deficiencies in the system. Through the test, users could operate multiple robots simultaneously with the multi-touch interface. Specifically, the test users successfully operated robots to gather in a specified area, avoid collision with obstacle and push boxes together.

論文要旨

ロボットに複雑なタスクをさせる場合、ユーザはまずロボットにすべきことを指示する。自律動作してタスクをこなすロボットでも、初めに指示を出すことが求められる。このため、ロボットの操作および教示インタフェースが必要とされている。この目的には、一般的に自然言語、ジョイスティックやその他のポインティングデバイスが用いられている。しかし、これらのインタフェースでは特に複数台のロボットを同時に操作することが困難である。

この不便を解消するため、我々は、複数台の移動型ロボットを操作するマルチタッチインタフェースを開発した。インタフェースには二通りの動作モードがある。一つは個々の指が別々のロボットを指示するモードで、もう一つは指でグローバルなベクトル場を編集し、全てのロボットが同一のベクトル場に従って動くモードである。

本論文では、このインタフェースとその実装およびユーザテストの結果について報告する。実験ではマルチタッチの二つの動作モードを比較した。結果から、動作モードはそれぞれ異なる特徴を持ち、各々の弱点を補い合う関係にあることが分かった。テストを通して、ユーザは複数台のロボットを同時に操作できた。具体的には、複数台ロボットを一か所に集めたり、障害物を回避しながら移動させたり、協調して箱を押させる作業を行った。

Acknowledgements

I would like to thank Associate Professor Takeo Igarashi for his pithy advice. I also thank my good neighbors who cared me when I worked rather in overload. Furthermore, I am grateful to all the members of Igarashi Laboratory for their cooperation.

Contents

Chapter 1

Introduction

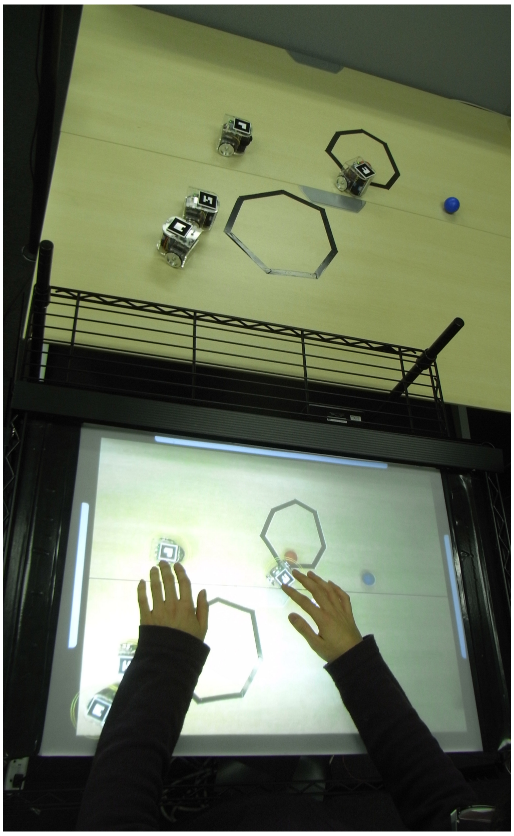

Figure 1.1: The proposed interface and a field on which robots move around.

All robots, including those that do their tasks autonomously, do not work without an instruction by users. We therefore need interfaces for giving instructions to them.

A single robot is normally controlled with joysticks, keyboards, and other pointing devices. However, with advances in robotics, variations of user interfaces for these purposes have become wider. For example, multimodal interfaces such as combination of hand gestures and speech for one assistant robot [1], and a portable interface using personal digital assistant (PDA) for mobile robots [2] have been proposed. This system allows users to navigate a robot with waypoints projected on the screen. Recently, user studies of intuitive interfaces [3] have been taken place, in which motor impaired users have a robot grab things with laser pointers and a touch-screen with buttons on it. Among these studies, it has been concluded that the invention of advanced user interfaces helps operating a single robot. Usability of robots largely depends on their user interface like personal computers, so studies of their interface are required toward prevalence of robots for home and entertainment use.

Handling tasks with multiple robots is also desirable, because they can do various tasks with greater efficiency than a single robot. However, multiple robots substantially increase amount of information exchange with their users who have to maintain situational awareness and continue operation. It often makes the manner of operation complex and difficult. Since users have a limitation in the capability of their attention, they cannot see too much information displayed either at the same time or time-multiplexed. Therefore, many user interfaces for operating a single robot do not work effectively. Designing interactions between people and multiple robots to achieve their effective cooperation has been a difficult research issue.

On the other hand, recent studies about Human Computer Interaction (HCI) have shown rapid progress. Multi-touch interaction is specifically getting popular with commercialization of the technology, and its key advantage unlike other input devices seems that its users can treat virtual objects as if they are directly touching the objects with their fingers or hands. When making multi-touch operations affect the real world, the users are expected to operate real objects very intuitively.

This paper proposes an interface using a multi-touch display to control multiple mobile robots simultaneously. The display shows a top-down view from a ceiling camera in real time. Users can control robots by two operating modes. One mode is named "Direct Operation" in which every contact surface directs its nearest robot on the view. In this mode, users move robots by dragging their icons. The other is named "Vector Field Operation" in which the view is virtually overlaid with a 2-dimensional vector field. All robots follow the field. In this mode, users manipulate the vector field by touching and passing their hands on the display. These two modes allow users to control robots intuitively, and help effective cooperation of people and robots.

Chapter 2

Related Work

2.1 Cooperation of People and Multiple Robots

Upon the cooperation of people and multiple robots, existing studies have tried to give robots some intelligence. This approach aims to allow for limited resources of people to care the situation and make orders continuously. Some have continued studies which only tell robots initial state and make them work autonomously [4]. Others like Fong et al. insisted that completely autonomous approaches are not yet feasible, and that robots should engage in dialogue with their users when required [5]. The paper indicates two factors for effective cooperation of people and robots. The first factor is that roles and responsibilities against tasks are clearly separated between users and robots. It is generally said that users should have responsibility for global tactics, and robots for local tasks. When the distinction between global and local tasks is unclear, problems occur for both the robot and the user. The second factor is that users can command robots easily as possible. Here we need richness of user interfaces cultivated in the field of HCI. Wang et al. conducted experiments and compared performance to accomplish tasks of robot teams with different degrees of autonomy [6]. They concluded that neither completely autonomous nor dependent on operators but "mixed initiative" teams did the best. In those teams, each robot works autonomously only when users are too busy to maintain its movement. Their result suggests that improvement of user interfaces to command robots in real time can help achieving better performance, because they may cut down users' busy time. Discussion of Driewer et al. describes how the user interface in human-robot teams should be [7]. They pointed out that in teams consisting of people, robots and their supervisor, the use of graphical user interfaces (GUI) greatly affect the performance of their tasks. As a whole, it can be said that user interfaces can play a considerable role in cooperation of people and multiple robots. This paper adopts this stance and tries to help achieving their better cooperation.

2.2 Multi-touch Display

Advances in technology have made many rich input devices. These days, specifically multi-touch display of such kind is very popular among the general public. Multi-touch technology itself was a concept dating back to the latter of 1990's like HoloWall of Matsushita and Rekimoto [8], but it has come to be popular through the commercialization and drop in cost of the technology. iPod touch of Apple is maybe the most affordable product with multi-touch display. Windows 7 of Microsoft officially declares support for multi-touch display devices. Among researchers, Han proposed the way to make a low-cost tabletop multi-touch display [9]. Tabletop systems with multi-touch capability are now provided by many companies such as Microsoft, Mitsubishi. Despite these advances in the technology, studies about its application have been progressed slowly in relative. The proposed system is a new example application on the border of HCI and Human Robot Interaction.

2.3 Top-down View

When we use a tabletop system, we usually look down on the surface. So, with a top-down view of some situation projected on the tabletop display, our lines of sight are simply extended through the surface into the situation. As a result, the projection looks very natural for many people. Another advantage of the top-down view is that the observers can see the situation at a glance. Thanks to these advantages, top-down view is very popular among entertainment applications.

When we operate multiple robots in real time with interfaces assuming their use as operating one robot (such as [2]), we usually have to switch our attention on every robot. A tabletop system with top-down view helps to solve this problem in a similar way of entertainment applications. There have been already user interfaces to command move of robots using top-down view, but many of them do not allow us to command robots directly and in real time on the view. A user interface which can directly command robots by drawing a sketch of the environment on a map of downward viewpoint exists [10], but it does not use any information from sensors getting a global top-down view like cameras set on the ceiling. "Direct Operation" mode proposed in this paper enables users control moves of each robot in real time by specifying the waypoints on a display.

2.4 Focus on the Field

Although users can maintain their situational awareness with the top-down view, they have to switch their attention among all robots when they command robots individually. It may yet be useful when users want to control each robot precisely. On the other hand, when the interactions between users and robots are designed so that users control robots as one swarm, there will be further less attention switches. One possible way to achieve this is to use the field as a proxy of all robots. Manipulating the field will affect all robots in this way.

Virtual Force Field [11] proposed in the dawn of collision avoidance studies used a global potential field to decide where robots should go locally. Each observed obstacles have some potential, in other words, areas around obstacles are positioned virtually high and difference in the height brings robots to a lower place. Regardless of obstacles, the potential field can be used to guide robots. A user interface that we can increase or decrease potential of the position we push [12] was developed and used for entertainment purposes such as video games. This interface can be used to operate real robots, but it produces little move in horizontal plane while robots mainly move horizontally. As a result, pushing the display seems not to be an enough interaction in order to make users feel like moving robots along their arms.

Instead, "Vector Field Operation" mode is proposed in this paper. It uses the motion of touched surface on the panel. Grids are made in the field to hold 2-dimensional vector information. When touched surface moves on the panel, grids in and near its track are affected to remember the direction of the motion. All robots decide in which direction to go according to the sum of vectors of grids near their position.

Chapter 3

Multi-touch Control Interface

The proposed interface is designed to allow its users to maintain their situational awareness easily and control movements of multiple robots intuitively. Robots are expected neither roles nor responsibilities more than their function of moving to some place where users intended. That is to say, roles for robots and users are clearly defined. Through the interaction design described below, we aim to satisfy the key factors for effective cooperation of people and robots.

3.1 Top-down view

A tabletop panel is used with a top-down view captured from a ceil-mounted camera. When our interface is used for home use or entertainment use, in most cases such as using robots in a room, physical position of the environment is static. Real time images of top-down view can be captured with cameras in such an environment. Even if setting cameras on the ceiling is difficult, it is possible to construct a map of top-down view gradually with the feedback information of sensors of robots. Main point is to use a top-down view of the environment with which users can see the global situation at once. Icons of detected robots are overlaid on the real image of them with almost the same size, and descriptions of them (texts of their name and status like "Stopped", "Rotating" or "Moving forward" etc.) are shown near the icons.

3.2 Mode change

There are two panels including buttons off the screen. To access these panels, there are handles at the left and right edge of the panel. By grabbing the left handle and pulling it to the center of the screen, we see two buttons labeled "Direct Operation" and "Vector Field Operation". We can change the mode with these buttons. The panel hid in the right edge has different functions in each mode and explained in the rest sections.

3.3 "Direct Operation" mode

The system is by default set to work in "Direct Operation" mode, in which users can control each robot individually. Operations available in the current implementation are shown below.

Drag When users touch and pass their fingers on the panel, the fingers' tracks are displayed on the screen and each track is assigned to a robot that is nearest to the start point of the track. Tracks work as directive of waypoints robots should follow.

Touch a robot icon Touching icons of robots on the panel will clear their waypoints. Touched robots immediately stops.

Clear all There is a panel including a button labeled "Clear all" off the screen. To access this panel, there is a handle at the right edge of the panel. By grabbing the handle and pulling it to the center of the screen, we can use the button to stop all robots at once.

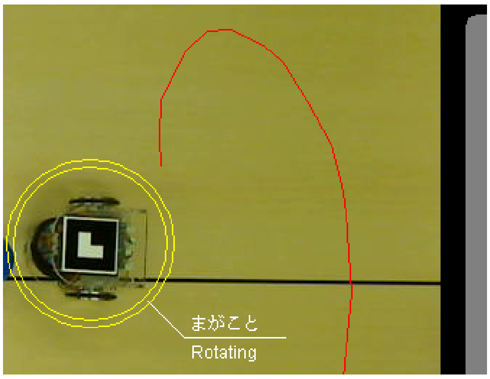

Figure 3.1: A screen capture of the system in "Direct Operation" mode. Waypoints to be visited are visualized with red lines.

3.4 "Vector Field Operation" mode

In "Vector Field Operation" mode, the vector field starts to hold its vector data when users pass their hands on it, but all the vectors shorten over time. In a figurative sense, all streams made by hands get thinner as time goes on. Operations available in the current implementation are shown below.

Drag When users touch and pass their hands on the panel, a virtual stream appears on the vector field and robots move according to the stream.

Touch Touching the panel without motion will clear vectors under the surface. Thus, we can stop a robot by touching an area where we predict it may go through.

Clear all By grabbing the right handle and pushing the button labeled "Clear all", all waypoints assigned to robots are cleared. All robots immediately stop.

Mix When users drag their fingers on existing streams, vector data near the streams are blended in proportion to distances against streams. Nearer streams affect stronger on the vectors.

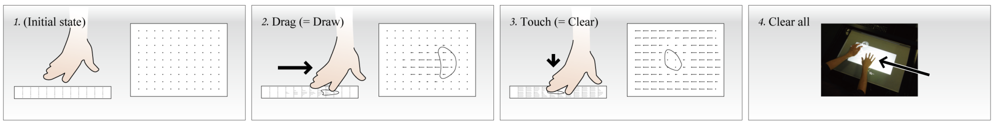

Figure 3.2: A list of statuses of the vector field and related operations

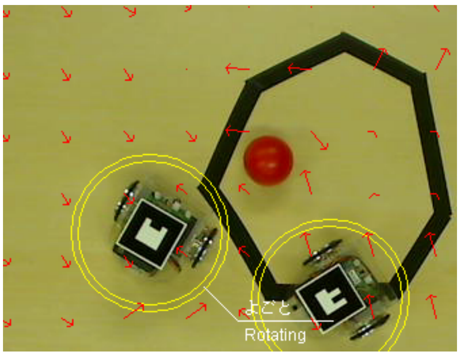

Figure 3.3: A screen capture of the system in "Vector Field Operation" mode. Vectors on grids are visualized with red arrows.

Chapter 4

Implementation

4.1 Hardware

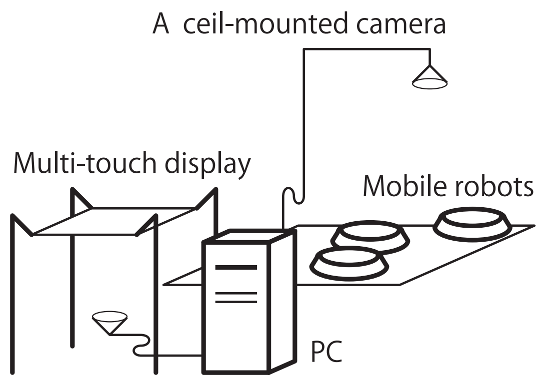

Figure 4.1: The overview of the hardware set up

The multi-touch interface adopts the low-cost method proposed by Han [9], using frustrated total internal reflection of infrared light in the acrylic panel. The shape of the touching surfaces can be detected by infrared camera set under the panel. A downward-pointing camera is set on the ceiling. Roomba robots by the iRobot Corporation and original small mobile robots developed at JST ERATO Igarashi Design Interface Project [1] are used.

[1] `http://designinterface.jp/`

4.2 Software

Robot Control The back-end system is built using the Java platform, and is checked to work properly on Mac OSX and Windows. Bluetooth links all types of robots to the computer with JSR-82 implementation for connection. The system control robots by sending four types of commands: go forward, rotate left, rotate right, and stop. All robots are represented by instances of a class named "Robot", and the class has member variables of two abstract classes named "RobotDriver" and "Worker".

Every type of robots has its own representing class extending RobotDriver like "RoombaDriver". The suitable driver class is instantiated in the constructor of a Robot instance, and used to communicate with the corresponding real robot via Bluetooth. RobotDriver class is designed to absorb the difference of command specification and performance of all types of robots. As a result, a robot capable of the four commands described above can be expressed as an instance of the Robot class. Status of a robot is distinguished by the name of a class extending "Worker" class whose instance is assigned to the robot. Each subclass of Worker represents a unit of task of a robot. Its instance runs as a thread and control the robot to accomplish the task by calling methods of the RobotDriver of the robot when needed. Four subclasses are used in the current implementation: Mover, Rotator, WaypointsTracer, VectorFieldWalker. Their roles are described below.

Mover This worker moves a corresponding robot to a certain destination in the field. It uses a Rotator instance inside to turn toward the destination, if an angle between the anterior direction and the destination direction is larger than a defined value (10 degrees).

Rotator This worker rotates a corresponding robot to a certain direction in the field.

WaypointsTracer This worker is used in "Direct Operation" mode. It makes a corresponding robot visit certain waypoints. It uses a Mover instance to visit each waypoint.

VectorFieldWorker This worker is used in "Vector Field Operation" mode. It calculates a vector at the position of a corresponding robot and decides how the robot should move. To be more precise, it decides the movement with the same logic as a Mover whose destination is where the head of the calculated vector points.

As shown above, some Workers call others inside their tasks. By defining each task as a class unit, a relatively complex task like WaypointsTracer can be described with combinations of simple tasks.

RobotDriver and Worker are designed to make the structure of the system clear and to separate the back-end system from the user interface programming. In addition, they enable easy extension of the system. For example, using new types of robots only need a new RobotDriver and adding a new operation mode may be achieved by making a new corresponding Worker. This extendibility would help the future work.

Image Processing Luminous blobs on the multi-touch display are visually detected using a marching square algorithm; and through calibration, we get position and size information of touched surfaces. In the calibration process, positions of blobs in the captured rectangle are projected into the screen coordinates. All shapes of blobs are approximated by ellipses which can be expressed by using the center position and lengths of major and minor axis.

Images from the ceiling-mounted camera are captured at 1/30 fps through QuickTime in Mac OSX or DirectShow in Windows. Positions of the robots are calculated through detecting markers in the captured images using Java implementation of ARToolKit [13]. Captured images work as both intuitive interface and a sensor in the system. Every time positions of robots are updated, how robots should move is calculated in instances of Worker subclasses, and if needed, commands are sent to robots wirelessly by instances of RobotDriver subclasses.

Direct Operation All robots have an array list of waypoints they are going to visit in "Direct Operation" mode. When luminous blobs are detected, the system finds out the nearest waypoint of any robots with each blob. If there is no waypoint near enough (nearer than 30cm in the real scale on the field in the current implementation), the position of a new blob becomes a first waypoint of the nearest robot. Otherwise, the position of a new blob is thought as a new waypoint following the nearest existent waypoint. If the followed waypoint is not the last element of an array list, the rest elements are discarded and the new waypoint is pushed to the list. A robot visits its waypoints and pops the visited from its list in sequence until the list becomes empty.

"Drag" operation means to manipulate array lists of waypoints. When users "touch a robot icon", a luminous blob is expected to appear near the waypoints of the intended robot by necessity. Then, with the algorithm described above, all waypoints farther from the robot than the touched position are cleared. Note that the touched position is inside the icon of the robot. It means that the robot has now no more waypoints needed to be visited; the robot is already at the final destination, so it stops. In fact, "Drag" and "Touch a robot icon" operations are implemented under the simple algorithm.

Vector Field Operation The screen is divided into a defined set of grids, and each grid holds 2-dimensional vector data. In the tested environment, the grid interval is 46 pixels, which is actually 15 cm long on the floor. Every time the information of luminous blobs is updated, their motion is tracked using a very simplified algorithm for optical flow. When luminous blobs are detected, the system finds out the nearest luminous blob in the previous captured image. If there found to be a blob nearer than a defined threshold value, the new and old blobs are recognized as the same, continuously moving blob. Otherwise, the new blob is neglected at this time, but its information would be used in the next time as an old blob. Tracked motion affects the existent vector field in the manner that grids directly under the touched surface are completely overwritten with the motion vector, and that those near the surface are blended with it in proportion to the distance against the center of the surface. Grids further than a defined distance (92 pixels i.e. 30 cm) are not affected. All vector data is shortened to a defined rate (98%) every time the information from the camera is updated. So, after neglected for a while, the vector field will convergent to the initial state.

Every robot moves to the direction in which sum of vectors held by near grids points. This addition uses a reverse manner as a luminous blob affects grids near it. That is, nearer grids affect strongly and further ones weakly in proportion to distance. Grids outside a certain circle whose center is the position of robot are neglected. In the current implementation, robots can only rotate or move forward. When the difference between the calculated and current direction is above a defined threshold ($\pm$ 10 degrees), the robot starts to rotate instead of going forward.

"Drag" and "Mix" operations are naturally achieved under the explained algorithm. In addition, when users simply touch the surface and do not move their hands, still luminous blobs are detected continuously in the touched areas. Motions of these blobs are little, so the grids near those areas come to hold almost zero vectors. As a result, robots on those areas stop. This is exactly how the "Touch" operation works. Three operations of "Vector Field Operation" modes are implemented under one algorithm like the case of "Direct Operation".

4.3 Result

The implemented system is designed to be able to execute both on Mac OSX and Windows platform. During the implementation, large part of development and debug was done on a Windows PC with Core2Duo E8400 processor and 2GB DDR2-SDRAM memory. A user test described in the next chapter was also conducted on this PC. Execution on the Mac OSX was tested on a MacBook Air with 1.6GHz Core2Duo and 2 GB DDR3-SDRAM memory.

Two web cameras are used to get a top-down view and to see the multi-touch panel. Their specification claims that they are capable to capture images at 30fps, but the frame rate often decreased to 15fps or less, maybe because of low-light intensity of the environment. Among all, this decrease seemed to be a bottle-neck of the performance of the user interface. For example, tracking fingers does not work well when the frame rate is too low. The frame rate partly relies on the parameter of the camera driver. Thus, parameter adjustment of the camera driver was required when using the system. Using cameras for professional use must improve the performance.

Despite the problem of frame rates, the system was very robust when kept running for over hours. It continued working until the battery of robots died. Bluetooth connections did not interference with each other when 7 robots (maximum number of connections by the Bluetooth specification) were connected simultaneously to the PC. To confirm the system working properly, tasks and applications listed below are put in execution. Some of them were conducted in the user test, too.

- A user has robots push a box simultaneously in "Direct Operation" and "Vector Field Operation" modes.
- A user has robots move around without collision with still obstacle on the field in both modes.
- A user has robots move around a person on the field in "Vector Field Operation" mode. When the person moves, the user has robots follow the person.
- Two users use the system at the same time, and played a game using "Vector Field Operation". The users have their own territory in the field. There is a red ball on the field, and each player competed with each other to keep the ball in their own territory for longer time during the game.

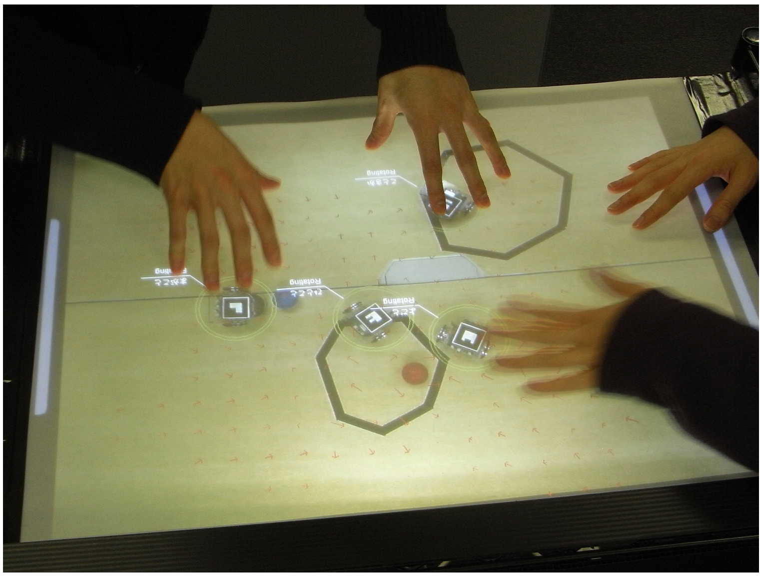

Figure 4.2: Two users are playing the game

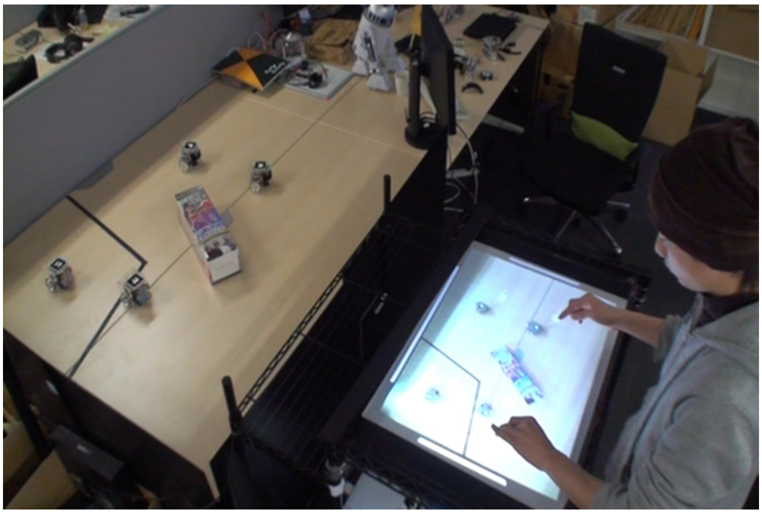

Figure 4.3: A user is controlling robots to avoid obstacle with "Direct Operation" mode. This is a part of the user test. Users are asked to accomplish three tasks with two operating modes separately and answered four questions in the test. Its detail is described in the next chapter.

Chapter 5

User Test

5.1 Design

A user test using the system was conducted. The aim of this test is to verify the usefulness and to figure out the characteristics of two operating modes: "Direct Operation" and "Vector Field Operation". The test did not compare these modes with operation using a tablet PC or a mouse. These devices work similar to a multi-touch display in "Direct Operation" mode, despite their limitation that they cannot command multiple robots at the same time. In other words, standard pointing devices are regarded to have less or at most the same usability of the multi-touch display. So, tests with them were abbreviated. Four users attended the test.

5.2 Method

Users who cannot use the multi-touch display well may not be considered as good testers, so they were at first required to be familiar with the display in large part. They were asked to touch the multi-touch display for a while. They played with a trivial program which draws touched areas with colored ellipses. After the practice to use the display, they are asked to try to accomplish three tasks with two operating modes. Through these tasks, four robots were always controlled simultaneously by users in real time. Each task was tried twice by all users, former with "Direct Operation" and the latter with "Vector Field Operation". They are not allowed to use two modes in one trial.

The first task was to gather robots that straggle at the initial state. All robots are required to go in the same defined area at the corner of the field. The second was to move robots from the one side of the field to the opposite side with an obstacle set at the center. Robots at the initial state were lined up in the left side. In this task, users were required to make robots avoid the obstacle and go to the other side. The third and final task was to make all robots push a box together. A box was set almost at the center of the field, and robots were lined up like the third phase.

After all these tasks, they answered a simple questionnaire including four questions. Three questions were about three tasks. Each question asks which operating mode ("Direct Operation" as D or "Vector Field Operation" as V) was easier to use in the task and why it is so. The final question is a free form in which users can write any comments on the system.

5.3 Result

For the first question, two of four users answered D and others V. At first glance, users seem to have different opinions. However, after reading the reason why they chose the answer, all of their impressions found to be consistent. Those who said D is good explained that when all robots get closer near the goal area, formation of robots has to be designed to avoid collision with each other; and that drawing up robots in some formation needs precise operation of each robot which can be only accomplished with D. Others who answered V said that making streams pouring to the destination area were very intuitive and useful. They do not care about the formation of the final state. In fact, their neglect is logically correct because the task did not require alignment of robots in the destination area. These results clarify the characteristics of two operating modes: V is suitable for ambiguous instructions which allow a large margin of error while D can conduct a delicate maneuver.

Answers to the second question were unified as V by all users. They insisted in chorus that V only has to make one stream around the obstacle while D has to specify similar waypoints of all robots a number of times.

The third task to make robots push a box together was apparently the most difficult among all the tasks, and three of the users answered D was better. Their preference on D is easily understood by the result from the first question. D can move each robot precisely, so users can try to make robots push one side of the box equally. As a result, they could push the center of gravity of the box and brought it. With V, they fail to push the center of gravity and have the box rotated for many times. Despite them, one user answered V was better. He indeed succeeded in pushing a box from one side of the field to the opposite side with drawing strokes on the display. His operation made a successful stream on the field, and robots were drifted by the stream, pushed the box simultaneously. He analyzed his success himself and said that he might succeed because he carefully planned where and how to draw a stream to foresee the movement of robots.

The last question was asking about the whole test, so the answers vary. A breakdown of the answers is shown below.

- Fingers ache pushing the surface. More sensitive display is needed.
- Grabbing the right handle to clear current situation was difficult.
- Using more robots may benefit V.
- V gives a very interesting interaction experience.

Related with the first answer, the FTIR multi-touch display has a weak point that its sensitivity is easily affected by the luminous environment. For example, it is difficult to use the display with bright sunlight. This should be surely improved in some technical way, but it is not a main topic in this paper. The second answer, a complaint about the GUI parts is a precise indication. The handles shown on the both sides of the display to grab the hidden panels have about 1cm width, and it may be too narrow to grab and pull. The width should be wider or the shape of the handles needs a review. The third answer is consistent with the result from the second question that D may need more work than V when the number of controlling robots increases. The fourth answer assures that the V gives users a novel experience of controlling robots.

Throughout the test, all users could accomplish all three tasks with at least V or D. It can be concluded that users can control movements of multiple mobile robots simultaneously with two operating modes. Some users were observed to command robots redundantly. They repeated the actions in order to confirm that their directions were certainly accepted by the system.

Chapter 6

Discussion and Future Work

6.1 Two Operating Modes

The user test revealed important characteristics of two operating modes. It became clear that "Vector Field Operation" allows users to have robots do ambiguous task intuitively, while "Direct Operation" enables to command each robot precisely. When whole robots are expected to accomplish a unique task as a swarm and each robot is not required to do its specific task, "Vector Field Operation" exceeds "Direct Operation" in performance. Since "Direct Operation" requires its users' work to be commensurate with size of the swarm, this difference becomes clearer when the size gets larger. The users of "Direct Operation" take its advantage when they want to operate only one robot despite the others. In regard to this case, "Vector Field Operation" does not work well. As a whole, both operating modes make up for each other's deficiencies in the system.

Difference of the characteristics are not limited to what kind of tasks they are appropriate for, but even includes what kind of functions they have. "Vector Field Operation" delivers unique movement of robots that cannot be achieved by "Direct Operation". For example, current implementation of the vector field can make robots go around a static loop like a circle for many times, by drawing a stream with its beginning and the end connected. This cannot be achieved by "Direct Operation" teaching waypoints which have finite length. Meanwhile, there found to be equally some limitations in "Vector Field Operation" mode. The vector field cannot bring robots on a path going across itself.

Since the two modes are completely exclusive in the current implementation, these differences may sometimes cause a trouble or inconvenience. For instance, when users are taking advantage of "Vector Field Operation", they might want one specific robot to move in an intended way. This kind of demand for local fix of the movement of some robots was certainly observed in the user test. To solve this problem, the implementation should integrate two operating modes. The system can be mainly based on ”Vector Field Operation”, with an optional function of ”Direct Operation” activated only when fingers start dragging on icons of the robots.

6.2 User Interface

In the current implementation, users can know whether their commands are accepted or not only by the change of status text beside robots. As a result, through the user test, they often could not recognize whether their commands are accepted or not and tended to command robots redundantly. It might cost additional machine resources or consume more wireless bandwidth compared with the least set of commands. More noticeable and comprehensible visual feedbacks will help users to be easy about operating robots. The system in the future work may show a notification balloon beside a robot only when some types of events about the robot meaningful for users are occurred.

In the box-pushing task of the user test, one user manipulated the vector field very well. His success can be thought only as an accident, but his explanation that he carefully designed the vector field suggests one assumption: he was good at foreseeing the movement of robots on a stream he is going to draw. The others might manipulate the vector field without enough forecast. According to this assumption, to make the system help users to forecast the movement of robots may achieve better performance. Showing particles virtually drifted on the vector field may work.

Along with the integration of two operating modes described in the previous section, the usability of the system can be further improved by adopting other user interfaces as exceptional operations on the basis of the vector field manipulation. Since ”Vector Field Operation” is not so good at establishing a delicate directive, the improvement to remove this weak point is required. Inspired by the user interface using a hand-drawn sketch to control robots [10], we may adopt a method which allows users to draw a virtual wall that cannot be crossed by robots. Erasing tool and menu buttons for changing modes should also be equipped.

Another possible approach is to define virtual objects on the field with positive potential which alienate robots. Users can drag and drop them working as if they were sheep dogs chasing mobile robots as sheep. This approach is similar to ours in that it makes users focus on a few virtual things without dividing capability of attention into each robot. Here we may combine the concept of boids [14]. It defines movement of robots bound by simple equations about their relative position. When users chase robots with a virtual sheep dog, robots as boids with a proper relational equation may succeed in escape without collision among themselves.

6.3 Applications

The interface can track locations of robots globally. In the future work, we may record and play the waypoints they passed. We can play a record only on the screen (in other words, without affecting real robots), where we can possibly seek time by choosing a visited waypoint instead of clicking a certain point on a normal seek bar. Applications for sweeping robots may allow users to register some of their favorite actions like cleaning only around a desk, avoid a trash-box, and so on, and play them whenever needed. Furthermore, for entertainment use, the system would allow a supervisor to see the field from the god view to make multiple robots interact with audiences in real time.

Chapter 7

Conclusion

We developed a multi-touch interface to control multiple mobile robots simultaneously with a top-down view from a ceiling camera. It has two operating modes: ”Direct Operation” mode which users directly draw waypoints of each robot, and ”Vector Field Operation” mode which users manipulate a vector field followed by all robots. The user test shows that ”Direct Operation” can conduct a delicate maneuver while ”Vector Field Operation” is suitable for ambiguous instructions which allow a large margin of error. As a whole, these modes make up for each other’s deficiencies. Users successfully controlled movement of multiple mobile robots simultaneously over the three tasks with this interface. Our study implies that enhanced HCI can offer a partial solution for the bottleneck of current HRI problem such that people have limited capability of attention. We are going to integrate two operating modes and extend the user interface to achieve more enhanced usability and accomplish more complex task using multiple robots.

References

[1] O. Rogalla, M. Ehrenmann, R. Zollner, R. Becher, and R. Dillmann. Using Gesture and Speech Control for Command a Robot Assistant. In *IEEE International Workshop on Robot and Human Interactive Communication*, pages 25–27, 2002.

[2] T. Fong, C. Thorpe, and B. Glass. PdaDriver: A Handheld System for Remote Driving. In *IEEE International Conference on Advanced Robotics 2003*. IEEE, 2003.

[3] Y.S. Choi, C.D. Anderson, J.D. Glass, and C.C. Kemp. Laser pointers and a touch screen: intuitive interfaces for autonomous mobile manipulation for the motor impaired. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, pages 225–232. ACM New York, NY, USA, 2008.

[4] D.T. Pham, M.H. Awadalla, and E.E. Eldukhri. Adaptive and cooperative mobile robots. *Proceedings of the Institution of Mechanical Engineers*, 221(3):279–292, 2007.

[5] T. Fong, C. Thorpe, and C. Baur. Multi-robot remote driving with collaborative control. *IEEE Transactions on Industrial Electronics*, 50(4):699–704, 2003.

[6] J. Wang and M. Lewis. Human control for cooperating robot teams. *ACM SIGCHI/SIGART Human-Robot Interaction*, pages 9–16, 2007.

[7] F. Driewer, M. Sauer, and K. Schilling. Discussion of Challenges for User Interfaces in Human-Robot Teams. In *Proceedings 3rd European Conference on Mobile Robots*, 2007.

[8] N. Matsushita and J. Rekimoto. HoloWall: designing a finger, hand, body, and object sensitive wall. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 209–210. ACM New York, NY, USA, 1997.

[9] J.Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118. ACM New York, NY, USA, 2005.

[10] M. Skubic, D. Anderson, S. Blisard, D. Perzanowski, and A. Schultz. Using a hand-drawn sketch to control a team of robots. *Autonomous Robots*, 22(4):399–410, 2007.

[11] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man and Cybernetics*, 19(5):1179–1187, 1989.

[12] K. Fukuchi and J. Rekimoto. Marble Market: Bimanual Interactive Game with a Body Shape Sensor. *Lecture notes in computer science*, 4740:374, 2007.

[13] H. Kato. ARToolKit : Library for Vision-based Augmented Reality. *Technical report of IEICE. PRMU*, 101(652):79–86, 2002.

[14] C.W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34. ACM Press New York, NY, USA, 1987.