Three Dimensional Tangible User Interface for Controlling a Robotic Team

Paul Lapides Ehud Sharlin Mario Costa Sousa
University of Calgary
2500 University Drive NW,
Calgary, Alberta, Canada, T2N 1N4
+1.403.220.6015
paul.lapides@gmail.com ehud@cpsc.ucalgary.ca mario@cpsc.ucalgary.ca

ABSTRACT
We describe a new method for controlling a group of robots in three-dimensional (3D) space using a tangible user interface called the 3D Tractus. Our interface maps the task space into an interactive 3D space, allowing a single user to intuitively monitor and control a group of robots. We present the use of the interface in controlling a group of virtual software bots and a physical Sony AIBO robot dog in an urban search and rescue environment involving a bomb hidden inside of a building. We also describe a comparative user study we performed where participants were asked to use both the 3D physical interface and a traditional 2D graphical user interface in order to try and demonstrate the benefits and drawbacks of each approach for HRI tasks.

Keywords
Human-robot interaction, tangible user interfaces, physical interfaces, interaction techniques, evaluation, robotic team control

1. INTRODUCTION
Efficient interfaces for controlling robots are still an open human-robot interaction (HRI) research problem. The challenge of controlling robots scales dramatically when considering tasks of shared interaction [8] between a single human operator and a team of multiple independent robots. Another more general challenge in human-computer interaction (HCI) is providing three-dimensional (3D) awareness of spatial interactive tasks. In this paper we address both challenges through the design, implementation and comparative evaluation of a new interface for controlling a group of robots in 3D space.

We designed a physical interface called the 3D Tractus that enables control of a group of robots in 3D space using a physical drawing board interaction metaphor. The 3D Tractus was designed with 3D spatial tangible user interfaces (TUIs) themes in mind. The interface allows direct physical mapping of a physical space, such as air space in the case of unmanned aerial vehicles (UAVs) or buildings in the case of urban search and rescue (USAR) robotic tasks. The user can interact in the 3D domain by physically sliding the 3D Tractus surface up and down in space. The 3D Tractus enables control of the robotic team on the interactive surface via simple sketch-based stylus actions. Overall, the 3D Tractus provides the user with intuitive spatial visualization of the task state as well as with tools for intuitively monitoring and distributing tasks between the robotic team members.

In this paper we describe the 3D Tractus-based robotic interface. The interface can be used to control of group of robots composed of independent AIBO robot dogs and virtual software entities. We describe our experimental test bed, a game-like 3D USAR environment in which a group of robots are controlled by a single human operator in a searching task. The user is asked to control the robotic team and explore a building in an attempt to find and defuse a hidden bomb. We use this test bed to pursue a fundamental question: would the physical 3D Tractus interface be more efficient for the team control task compared to a similar non-physical graphical user interface (GUI)?

In order to conduct a comparative study we mapped two interfaces to the experimental environment: the first is the Tractus TUI and the second is a Window-Icon-Menu-Pointer (WIMP) slider. This paper discusses our experimental design, the pilot study phase and a full study we performed comparing the two interfaces.

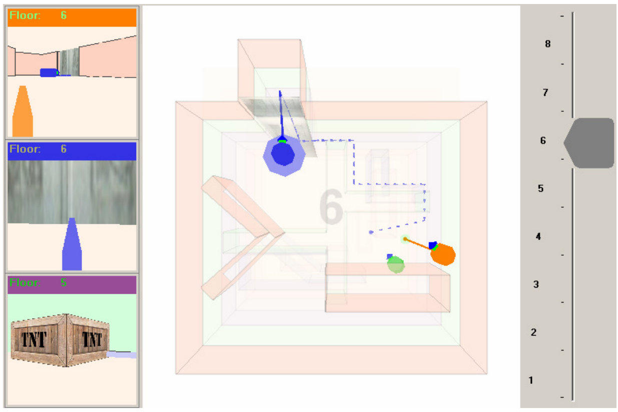

Figure 1: Robots can move in 3D through a building.

Our contribution in this paper is twofold: the design of a new physical TUI for 3D spatial control of a group of robots by a single user; and a user study comparing the interaction quality of the physical TUI to a non-physical GUI-based parallel interface in a simulated USAR-like environment.

2. RELATED WORK

Our work relates to the fundamental challenge of controlling of a team of independent robots by a single user. Following Yanco and Drury's taxonomy [8] our implementation can be useful for tasks with high amount of user intervention and with one human controlling a group of multiple independent robots.

Several previous PDA-based sketch-based interface systems for robot control have been proposed and can be categorized as relying on either quantitative [1-3] or qualitative [4-6] 2D maps.

Recently, Skubic et al [7] proposed a sketch-based interface to control multiple robots using a qualitative, hand-drawn map. The paper proposes a single user, Tablet PC sketch-based interface for controlling a team of robots along a sketched 2D route map. In their system, the user sketches a qualitative map of the environment describing the scene, and then navigation gestures are sketched to direct the robots. The interface also displays the robot's sensor feedback and allows the synchronization of robot locations. A usability study was conducted in two groups, one controlling the robots in an unaltered environment and the other group in a slightly altered environment. Most users validated positively the sketch-based interface for guiding mobile robots.

We designed the 3D Tractus as a TUI. TUIs are interfaces that couple digital information and function with physical artifacts [9]. Good spatial TUIs will provide intuitive spatial mappings between the physical object and the task, unification of input and output spaces and support of trial and error activity [10].

The 3D Tractus is a simple TUI which was originally designed for general 3D sketch-based interaction and for plant modeling [12, 11]. The user can move the 3D Tractus vertically up and down while sketching on a tablet PC placed on its surface. The 3D Tractus height is being tracked using a simple sensor and the stylus surface position is tracked through a tablet PC or any other touch sensitive surface interface. The 3D Tractus was designed to support direct mapping between its physical space to the task virtual space.

As far as we know this work is the first to explore the use of 3D sketch based spatial TUIs in human-robot interaction. This is also the first attempt to empirically compare TUIs effectiveness in a practical HRI control task to a non-TUIs control mechanism.

3. SYSTEM DESCRIPTION

3.1 Overview

We wrote an application that is used to control a group of robots in 3D from a high-level perspective. The user takes the role of team leader and gives macro instructions to the group, overseeing the overall progress of the robots without bothering themselves with micro details like location and guidance.

We chose to orient the application as a game to demonstrate the power of our system and to make it fun to use. A group of robots are dispatched into a building and are hunting for a bomb hidden inside. Robots are fitted with cameras that show their surroundings on the screen. The user must pay attention to these video feeds and once the bomb has been sighted they must guide the robots to the bomb in order to defuse it.

In the first iteration of the game the entire scenario was simulated on the computer, like a traditional video game. The building is an arbitrary arrangement of floors stored as a file with special locations for an elevator and bomb. The robots are also virtual entities and obey physical laws as real robots would. For example, they cannot move through walls and have realistic speeds for locomotion. Robots can only perform simple actions like turning and moving forward.

In the second iteration of the game, we integrated a Sony AIBO ERS-7M2 robotic dog into the system to show that our interface can be used to control real robots and not just simulated entities. The AIBO satisfied the minimum requirements for our environment because it can turn, move forward, and has a video camera that is used to show its surroundings. The AIBO robot is only a working concept as we currently have not implemented any form of tracking. Unlike the virtual entities, the system is not aware of the AIBO's exact position and orientation, meaning that it is behaving as an open-loop system. The AIBO will go off track after a period of use and must be "recalibrated" by aligning the real robot with its virtual representation.

The user gives instructions to the robots by specifying a place in the building they should move to. The building has several floors so the user must be able to move up and down through the building to access all the floors. We implemented two interfaces that give the user the 3D interaction abilities needed to move vertically throughout the building. The first interface is a traditional WIMP slider where the vertical position of the slider determines what floor of the building the user is currently viewing. The second interface is a TUI called the 3D Tractus. It is a movable table surface with a tablet PC running the application lying on top. The user slides the table up and down physically changing their viewpoint inside the building and clicking on the surface to issue robot tasks.

The application can run on any Windows PC and requires only a mouse (or stylus) with left-click functionality. We have combined the output (screen) and input (mouse cursor) context spaces into a single space by running the application on a Toshiba Portege M200 tablet PC and using the stylus to click the screen. It is written in C# using the CsGL OpenGL wrapper library.

3.2 Controlling Robots

The game has three important areas on the screen shown in Figure 1. In the middle is a top-down view of the building showing the floor the user is currently viewing. This is where the robots' locations are shown and instructions are given by clicking on the floor. On the left are the video feeds from each robot in the building. Each video feed has a color bar corresponding to the robot the video is coming from, as well as a number showing what floor the robot is on. The user must watch these feeds in order to identify the bomb when a robot passes near it. Finally, on the right is the vertical slider that controls the user's position in the building. In the slider mode the user controls the slider by directly dragging it up and down and in the Tractus mode the slider is

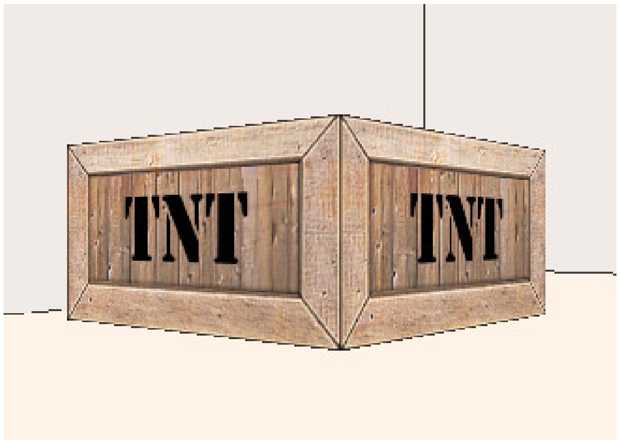

Figure 2: The bomb is represented by a box of TNT.

controlled by the physical height of the Tractus (when the Tractus is moved up, the slider will move up, etc.)

Robots are selected and deselected by clicking on them once. Once a robot is selected, a place on the floor is clicked and the robot automatically moves to that location. This is called a task. The user can keep queuing tasks after the first task is issued and the robot will move to the locations in the order they were given. The user can clear the robot's queued tasks by reselecting the robot (clicking on the robot again). By selecting more than one robot before issuing tasks, multiple robots will move in unison.

Most robots use wheels for locomotion and are not suited for going up stairs. Even legged robots have challenges with climbing stairs. Most buildings are equipped with elevators, and this is a convenient solution for allowing robots to move vertically through the building. To move between floors, the robot is selected as before and the user changes floors using either the slider or the Tractus. Once the desired floor is reached, the user clicks somewhere on the floor and a task is issued like normal. The robot will automatically find the elevator, move vertically to the target floor, and move to the destination.

The robots must find a single bomb that is represented by a wooden crate labeled "TNT" (Figure 2). This object easily stands out from the building and is easy to spot on the robots' video feeds. A robot does not change its behavior when the bomb is in sight but continues following the tasks that were assigned. It is the user's responsibility to navigate the robot to the bomb once they discover where it is hidden. If a robot is within a small distance of the bomb (1 meter in comparative terms) it will automatically stop and begin defusing the bomb. The color bar above the robot's video feed will begin blinking to notify the user of the robot's success and the robot will not respond to any more commands. Once the bomb is located, the user must guide all the robots to the bomb. Only when all the robots dispatched in the building are defusing does the bomb become fully disarmed.

3.3 Automatic Guidance

Our system is made to let the user manage from a high-level role and free them from micromanaging each robot. In order to accomplish this, the robots must be intelligent enough to navigate through moderately complex environments to reach their issued destinations. Floors are rarely just rectangular rooms but are full

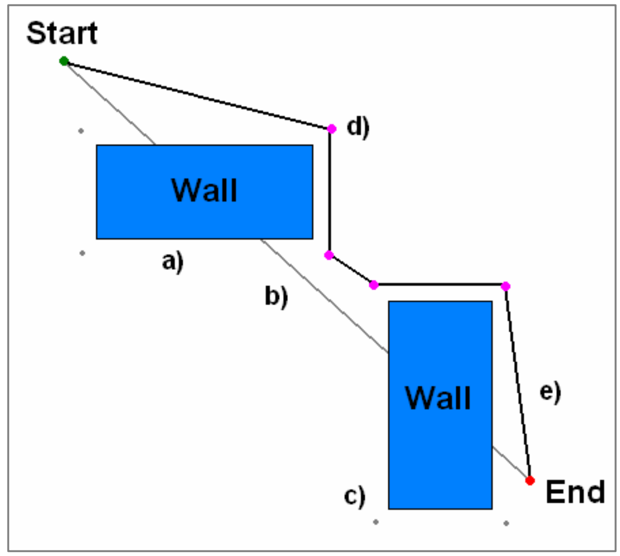

Figure 3: Guidance algorithm. a) Obstacles, b) Proposed path (occluded by walls), c) Margin points for walls, d) Intermediate tasks (margin points), e) New path created for the robot to follow.

of objects like chairs, desks, and other walls. For simplicity, the only kind of obstacle in the game is a wall, though other objects can easily be added. The robots have a basic AI to guarantee that they can reach any point on an arbitrary floor.

The guidance system is used when a given task does not have an unobstructed line from the start to the end; in other words, there is a wall in the way of the robot. The algorithm, briefly shown in Figure 3, creates intermediary tasks around the corners of the wall. The algorithm is recursive, so it is able to handle an unlimited number of obstacles. Due to complexity, the algorithm is not optimized to give the shortest possible route. It does not analyze the video feed and identify walls or use any kind of computer vision. The floor plan of the building is known a priori and for this reason the algorithm uses data directly from the layout of the walls.

3.4 Integrating Real Robots

The robots described so far are virtual entities handled entirely by the computer. The virtual robots only have to rotate, move forward, and provide a video feed, all of which is handled by the computer. As long as these three requirements are satisfied, physical robots can be integrated into the game and be controlled with the same interaction metaphors are before.

To demonstrate this feature, we programmed an AIBO robot to be controlled wirelessly by the game. The computer sends rotation and walking commands to the dog and in return the AIBO sends back JPEG frames from its built in webcam. There is currently no positional feedback that updates the AIBO's virtual avatar in the game. In other words, this is an open-loop system and there is no way for the game to know where the dog actually is. As far as the game is concerned, the AIBO entity is just another virtual robot and the game is in control of moving and guiding it on the screen (Figure 4).

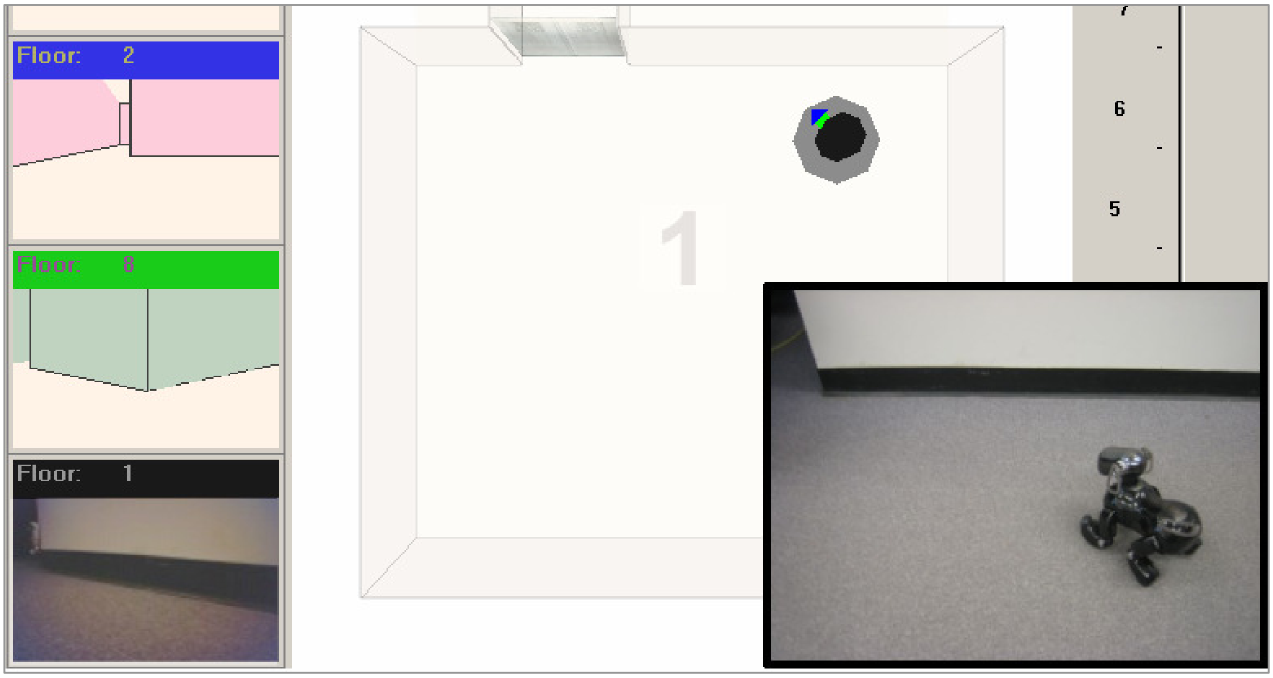

Figure 4: An AIBO dog is integrated into the system and has a real video feed, shown on the bottom left.

The turning and moving speeds of the virtual entity representing the AIBO have been calibrated to approximately the same speeds as the AIBO's movement. This means that the positions of the AIBO and its virtual avatar are consistent for the first few minutes of the game, but eventually deteriorate because the AIBO's movement is not precise.

3.5 Visualization

Several visualization cues are used in the top-down view of the game. The floors of the building are transparent allowing the user to see the floors below. This is done to show more information about the locations and tasks of the robots below the current floor. In the case of the Tractus, this joins the virtual and physical spaces by conveying the notion that there is actually something below the current level of the 3D Tractus's surface. The building view is top-down so it would not be logical to show the floors above the current height.

Making the floors transparent carries the risk of overloading the user with too much information and cluttering the screen. The transparency is controlled by a non-linear gradient, so floors just below the current height are still visible while floors far away are completely hidden (Figure 5). As well, the displayed size of the robot entities follows this gradient. The further away from the current height a robot gets, the smaller it is drawn so the user is not confused about what robots are on the current floor. Robots are highlighted and start to pulsate when they are selected to clearly distinguish then from unselected entities.

By far, the task lines have the highest potential to clutter the screen. If there are many lines, they will block objects behind them and confuse the user about what robot is going where. It must also be clear in what direction the task line is pointing and how far it is in the robot's execution queue. The first visualization that is applied to the task lines is the same as the floor transparency. The further lines get away from the current height the less opaque they become, until they eventually disappear at a certain distance. This prevents the view from being cluttered by task lines that are far below the current height. The second visualization, shown in Figure 6, is that lines are dashed more sparsely when they are further in the execution queue to show that the task is not being executed soon. The dashes are arrow shaped to show the direction of the task.

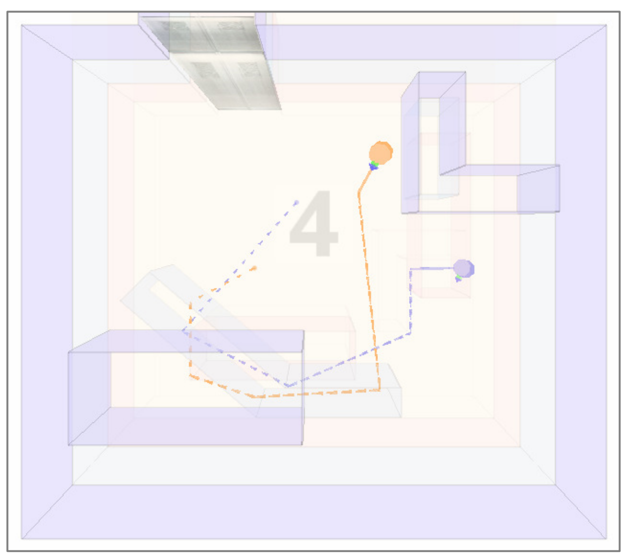

Figure 5: Transparent floors and tasks.

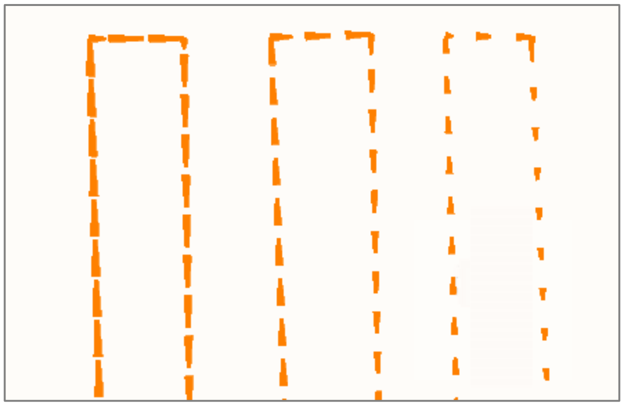

Figure 6: Task lines are sparsely dashed to show when they will be executed.

4. INTERFACES

Two different interfaces are used to the control the vertical movement in the game, enabling robots to move in 3D. The first is a traditional WIMP slider and the second is a tangible device called the 3D Tractus.

4.1 Slider

A vertical slider is shown on the right side of the screen. In this interaction mode, the user drags the slider up and down with the mouse to control his height in the building. This slider differs slightly from the standard Windows slider because the sliding widget is significantly enlarged. This makes it easier to click when the user wants to slide quickly.

4.2 3D Tractus

The 3D Tractus is a simple mechanical device consisting of a table surface that slides up and down on four vertical tracks. The frame, shown in Figure 7, is constructed from 80/20 Industrial Erector Set aluminum bars and can be easily assembled with a screwdriver. The application runs on a Toshiba Portege M200 tablet PC that lies on top of the table surface. The user moves the table surface up and down using their spare hand while either sitting or standing. A counterweight system similar to those used in elevators keeps the table surface balanced so the user does not have to support the weight of the tablet with their arm and makes it easier to move the surface up and down. The maximum vertical range of the surface is 18cm.

A string potentiometer measures the height of the table by increasing the voltage on its output when the string is extended and decreasing the voltage when retracted. An Ontrak ADU100 Data Acquisition Device (analog/digital converter) converts the voltage to a 16-bit unsigned integer. The ADU100 interfaces with Windows via USB and the height of the table surface is sampled in real time.

Because the user is looking down on the screen of the tablet, the top-down view of the building in the game combines the physical and virtual spaces giving the appearance that the building exists in the physical world.

5. USER STUDY

A pilot study followed by a user study was conducted comparing the effectiveness of the slider interaction mode to the Tractus interaction mode. As part of the study, we collected data about the participants' expertise in robotics and video games as well as their overall preference. Participants were recruited by sending notices to mailing lists in the Computer Science and Electrical Engineering departments.

5.1 Experimental Design

A pilot study was conducted with 6 participants who were experienced with computers and had some experience with robots. The pilot study was used to find bugs, adjust the experiment for time, and balance the study by making each of the bomb locations approximately the same difficulty to find.

Participants were explained the purpose of the study and shown the interface. They received training on how to move vertically in the building using either the slider or the Tractus, how to control the robots, and how to defuse the bomb. They were then asked to defuse a bomb in a short training exercise involving two robots called Orange and Navy (corresponding to their colors) and a two storey building with various obstacles. The bomb was hidden behind a wall on the 2nd floor. Once the user was comfortable with the game, the actual experiment was explained to them.

The user performed three trials with the bomb hidden in a different place each time. Users controlled three robots (Orange, Navy, and Forest) in an eight storey building. The Orange and Navy robots started on the first floor and Forest started on the eighth floor. The user guided the robots inside the building until the bomb was located and defused by moving all three robots close to the bomb, at which point the trial is complete. The bomb is hidden on the 4th floor for trial 1 and different places on the 5th floor for trials 2 and 3. For each trial the bomb is hidden behind a wall, forcing the user to explore each floor to be sure that they have not missed the bomb.

Once the three trials were complete, the user was trained to use the other interface. If they first used the slider they would be shown how to use the Tractus, and vice versa. The user was then asked to defuse a bomb in the same two storey building as in the first training exercise, with the same robots. The bomb was hidden

Figure 7: The 3D Tractus frame and counter-weight.

behind a wall in a different place on the 2nd floor. Next the user did three more trials with the same eight storey building and the same three robots as in the first three trials. The bomb was hidden on the 6th floor for trial 1, the 4th floor for trial 2, and the 5th floor for trial 3, all behind walls.

After the last three trials were completed, the user filled in a questionnaire that asked them to rate their understanding and performance in the game, as well as pick an overall favorite interaction mode (slider or Tractus).

As participants play the game, they learn the layout of the building, which certainly influences their performance by allowing them to better plan where to send robots and by giving them general practice. We avoided bias to either of the interaction modes by counter-balancing the interface order. Half of the participants first used the slider and half first used the Tractus. The positions of the bombs remained the same regardless of what mode was used first. That is, on the same trial number (1 to 6), a participant using the slider first will find the bomb in the same place as a participant using the Tractus first.

As the user searches for the bomb, the game logs all actions that the user performs. Everything from selecting robots, to changing floors, to giving tasks is logged in a file with timestamps and other attributes. The files are later analyzed by a custom program that extracts data for each participant on a per trial basis. For example, numbers are extracted for the time it takes to defuse the bomb, the number of robot selections made, the number of tasks given, etc.

5.2 Results

Twenty participants were recruited for the full study and the average age was 22.65 years with a standard deviation of 3.11 years. Most participants were students and had a lot of computer experience and moderate video game experience. Participants were compensated $10 for their time. The running time of the study was from 45 to 60 minutes.

Most participants had no difficulties with the system and managed to find and defuse the bomb in about 5 minutes, regardless of where the bomb was hidden. Several participants did not notice the bomb when it appeared on the robots' video feeds and subsequently had to re-explore part or all the building. The fastest time a bomb was found was 59 seconds and the longest was 669 seconds (11 minutes). Over all the trials and participants, the bomb was found on average in 221 seconds, with a standard deviation of 88 seconds.

The main comparison between the two interfaces is the time it takes for the bomb to be found. Figure 8 shows the time in seconds it took for the participants to find the bomb in four sets of three trials, as before. At first the Tractus fares very poorly compared with the slider, but the Tractus and slider perform the same by the end of the third trial when the user has had more training with the system.

It was found from the questionnaire that exactly half the participants prefer using the slider and half prefer the Tractus. Of the ones that prefer the slider, 50% actually performed the tasks faster when using the Tractus. It is interesting to mention that 55% of the participants performed better with the device that they did not prefer, showing no correlation between performance and perceived preference.

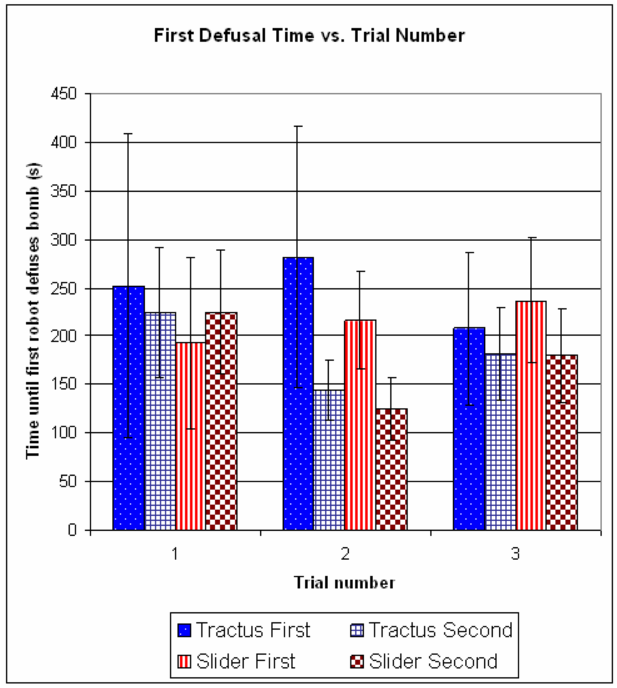

Figure 8: The time until the first robot in the trial finds the bomb and begins defusing against four sets of trials.

6. DISCUSSION

It should be reiterated that the purpose of the study was not to test the graphical interface but to compare the slider and Tractus against each other. The interface was purposefully crippled (for example, no audio feedback) in order to encourage people to use the slider and Tractus more. This study gave us an opportunity to look at how people used our system and how we can improve it in the future.

Three types of strategies were observed when the participants controlled the robots. Some users chose to micromanage by giving each robot only one or two tasks at a time and supervising its movement. The user's entire attention was focused on a single robot and this left the other two robots idle most of the time. This type of strategy was inefficient and did not result in very fast search times. On the other extreme, some users tasked the robots to move through every possible area of floor space, relying on the robot's feature of automatically stopping when it gets close to the bomb instead of paying attention to the video feeds themselves. As well, the robots were tasked to many floors at once and the user sat back and watched as the robots executed their instructions. Using this strategy, the user does not play a dynamic role in supervising the robots and does not give the optimum route for the robots by making them scan the entire floor. Users sometimes missed the bomb and had to start searching from the beginning. The largest majority of the users issued many tasks to the robots, usually assigning one robot to a single floor and waiting until the robot was finished searching. The robot was assigned new tasks on an unsearched floor until the bomb was found. This method was the fastest and most efficient because the user watched the video feeds and did not send the robot all over the floor.

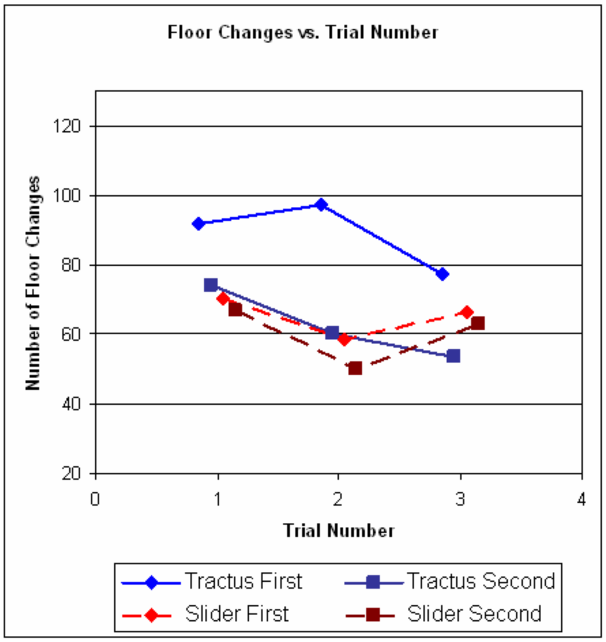

Figure 9: Number of floor changes against trial number for four sets of trials.

The participants slowly learned how to effectively use both the slider and the Tractus as they had more practice from the trials. The number of floor changes the users does is a measure that can reflect on how comfortable the user is with the device. If participants change floors too often it can mean that they are unsure of where they are in the building or have trouble moving to the floor they want. Figure 9 shows the number of floor changes for four sets of three trials each. The sets are organized by which interface is tested first (slider or Tractus). Participants that use the Tractus first change floors more often than participants that use the Tractus later. This suggests that the user is unfamiliar with the layout of the building and is still learning how to organize their strategies. The number of floor changes steadily decreases when the participants used the Tractus in the second half of the experiment, while the slider sets don't show an obvious decrease. This suggests that the user is more efficient with the Tractus after having sufficient practice with the device and with the environment.

Participants agreed that they were comfortable while using the Tractus based on an average of 4.4 on a 5-point Likert scale [13] with 1 being strongly disagree (to being comfortable) and 5 being strongly agree. Averages of 4.4 and 4.35 on a 5-point Likert scale (1 is strongly disagree, 5 is strongly agree) were found when users where asked if they understand where the robots are in the slider and Tractus trials, respectively, suggesting that both interfaces are equally clear to the user.

A seemingly important reason for participants to prefer the Tractus is because they could use their free hand to change floors instead of having to change context spaces between the robots and the slider. Several participants suggested a physical slider on the side of the tablet PC be used to change floors and that an elaborate device like the Tractus is not necessary. A few participants said that keeping track of all the robots and their video feeds was overwhelming and that they were prone to make mistakes and not notice the bomb as easily. Other suggestions were made like having a button to deselect all the robots, changing the projection from perspective to orthogonal, showing floors above the current height, and giving the user audio cues when important events occurred (such as the bomb being defused).

Several participants encountered a minor problem when selecting robots. Users would properly select and issue tasks to a robot and then move onto the next robot without deselecting the previous one. The system is designed to operate this way; when the new robot is selected, the old robot is automatically deselected. However, some users were in a rush to select the new robot and "missed" by not clicking on the robot but instead on the floor nearby. The system interprets this as another task being issued and the unwanted task is queued to the old robot. This problem is at best an annoyance because the unwanted task is placed at the very end of the execution queue and does not interfere with the tasks already assigned. The user could tackle this by simply remember that the robot has another task that it will complete, which is something few participants identified. The majority of the users panicked when missing clicks and wasted time clearing and reassigning all the tasks because of an unwanted task. A larger click area around the robots can be used to solve this problem.

7. FUTURE WORK

In the short term we are planning to follow up on the various user comments in order to improve our robotic team control mechanisms. We would like to extend the Tractus approach to various practical robotic applications: one approach would be to include tracking mechanisms that will allow us to send a group of real robots through a physical building using the elevator system to move between floors. Such an application can arguably be useful for security or USAR tasks although it is doubtful how well tracking mechanism will work in realistic disaster or war zones. Another direction is to use the Tractus interface for playing collaborative and realistic games with robots in a physical building environment. Robots can be sent as surrogates for the player, looking for various objects or other players.

In the long term, we are hoping to pursue mappings of the Tractus interaction paradigm to monitoring and controlling a group of airborne robots, such as UAVs. We believe that a Tractus, or a Tractus-like interface can be effective in controlling airspace routes and tasks of a group of robots that can flexibly change their 3D position.

8. CONCLUSION

We presented a TUI-based 3D interface enabling a single user to monitor and control a team of independent robots in spatial tasks. By mapping the 3D task to a simple physical interface we implemented an intuitive interaction technique that allows the user to explore 3D space by simply moving the interface up and down, following a drawing board metaphor. We demonstrated the potential of our interface in an USAR-like game using real and simulated robots and designed a study that compared the effectiveness of our 3D Tractus TUI approach to a non-TUI interface. While we believe our implementation and study are valuable, and while qualitative results were positive, we would like to stress that our current quantitative results are inconclusive in suggesting whether our TUI approach is "better" than the non-TUI one, or not.

9. REFERENCES

[1] Perzanowski, D., Schultz, A.C., Adams, W., Marsh, E., and Bugajska, M. 2001. Building a multimodal human-robot interface. IEEE Intelligent Systems, 16(1):16–21.

[2] Lundberg, C., Barck-Holst, C., Folkeson, J., and Christensen, H.I. 2003. PDA interface for a field robot. In Proc. 2003 IEEE/RSJ Intl. Conf. Intelligent Robots and Systems, Las Vegas, NV, pp. 2882–2887.

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

[4] Chronis, G. and Skubic, M. 2004. Robot navigation using qualitative landmark states from sketched route maps. In Proc. IEEE Intl. Conf. Robotics and Automation, New Orleans, LA, pp. 1530–1535.

[5] Freksa, C., Moratz, R., and Barkowsky, T. 2000. Schematic maps for robot navigation. In C. Freksa, W. Brauer, C. Habel, and K. Wender (eds.), Spatial Cognition II: Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications, Berlin: Springer, pp. 100–114.

[6] Setalaphruk, V., Ueno, A., Kume, I., and Kono, Y. 2003. Robot navigation in corridor environments using a sketch floor map. In Proc. IEEE Intl. Symp. Computation Intelligence in Robotics and Automation, Kobe, Japan, pp. 552– 557.

[7] Skubic, M., Anderson, D., Blisard, S., Perzanowski, D., and Schultz, A., Using a hand-drawn sketch to control a team of robots, In Auton. Robots, 22 (4), 399—410, 2007

[8] Yanco, H.A., Drury, J. 2004. Classifying human-robot interaction: an updated taxonomy. In 2004 IEEE International Conference on Systems, Man and Cybernetics, Vol. 3, 2841-2846

[9] Ullmer, B., and Ishii, H. 2000. Emerging Frameworks for Tangible User Interfaces, in IBM Systems Journal, Vol. 9, Nos. 3&4, 2000, 915-931.

[10] Sharlin, E., Watson, B. A., Kitamura, Y., Kishino, F., and Itoh, Y. 2004. On Tangible User Interfaces, Humans and Spatiality. In Personal and Ubiquitous Computing, special issue on "Tangible Interfaces in Perspective", http://delivery.acm.org/10.1145/1030000/1023818/40080338.pdf?key1=1023818&key2=2990849811&coll=GUIDE&dl=GUIDE&CFID=29345089&CFTOKEN=67796707

[11] Streit, L., Lapides, P., Costa Sousa, M., and Sharlin, E., 2006. Modeling Plant Variations through 3D Interactive Sketches. In 3rd Eurographics Workshop on Sketch-Based Interfaces and Modeling, SBIM 2006, Vienna, Austria.

[12] Lapides, P., Sharlin, E., Costa Sousa, M., and Streit, L. 2006. The 3D Tractus: A Three-Dimensional Drawing Board. In Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TableTop '06), Adelaide, Australia.

[13] Likert, R. 1932. A technique for the measurement of attitudes. In Archives of Psychology, 140:5–53.