TO BE PRESENTED AT THE 2012 IEEE/RSJ INT. CONF. ON INTELLIGENT ROBOTS AND SYSTEMS - IROS'12, OCT., 2012.

A Collaborative Control System for Telepresence Robots

Douglas G. Macharet and Dinei A. Florencio

Abstract— **Interest in telepresence robots is at an all time high, and several companies are already commercializing early or basic versions. There seems to be a huge potential for their use in professional applications, where they can help address some of the challenges companies have found in integrating a geographically distributed work force. However, teleoperation of these robots is typically a difficult task. This difficulty can be attributed to limitations on the information provided to the operator and to communication delay and failures. This may compromise the safety of the people and of the robot during its navigation through the environment. Most commercial systems currently control this risk by reducing size and weight of their robots. Research effort in addressing this problem is generally based on "assisted driving", which typically adds a "collision avoidance" layer, limiting or avoiding movements that would lead to a collision. In this article, we bring assisted driving to a new level, by introducing concepts from collaborative driving to telepresence robots. More specifically, we use the input from the operator as a general guidance to the target direction, then couple that with a variable degree of autonomy to the robot, depending on the task and the environment. Previous work has shown collision avoidance makes operation easier and reduce the number of collisions. In addition (and in contrast to traditional collision avoidance systems), our approach also reduces the time required to complete a circuit, making navigation easier, safer, and faster. The methodology was evaluated through a controlled user study (N=18). Results show that the use of the proposed collaborative control helped reduce the number of collisions (none in most cases) and also decreased the time to complete the designated task.**

I. INTRODUCTION

The way people interact and collaborate is continuously changing. In the business environment, one of the key factors currently driving this change is the increasing presence of geographically distributed working groups. This has caused a enormous growth in deployment of commercial telepresence systems and other remote collaboration tools. However, remote collaboration is still very far from satisfactory. Thus, research targeted at developing new means to improve collaboration between non-collocated groups has received significant attention.

Research on remote collaboration and telepresence spreads over many different areas. One particular type of device is, however, of marked interest to the robotics community: telepresence robots. Indeed, many in the robotics community are quite excited with the recent introduction of a number of commercial telepresence robots [1], [2], [3], [4], [5], with the potential to quickly become a significant market.

Although the concept of telepresence robots is very broad, systems used for this purpose mostly consist of teleoperated mobile robots, with the addition of particular elements targeted at communication (e.g., video displays, high quality microphones, etc). The main objective of these systems is to represent a human operator (remotely located), giving the person the ability to actively interact with the environment where the robot is placed.

While much progress has been made, several challenges are still present in the development and use of telepresence robots. Among the main difficulties we highlight the restricted amount of information the operator receives from the environment, and communication problems [6] that often occur between the control center and the robot (since wireless networks are very susceptible to failure). Indeed, remote operation of a (potentially heavy) machine may compromise the safety of people and of the robot during its navigation through the environment. For most commercial systems this is a very serious concern, as the company's future may be just a lawsuit away. Indeed, most commercial offerings currently control this risk by reducing size and weight of their robots [1], [2]. At a research level, we can look a few years ahead (and be a bit more courageous), and this is generally done by "assisted driving", which typically adds a "collision avoidance" layer, limiting or avoiding movements that would lead to a collision. However, this traditional approach — while helpful — is not the definitive solution. In particular, the task of controlling the robot requires a certain amount of attention from the operator in order to avoid accidents. With traditional assisted driving, the attention required to navigate is still significantly high, and will often interfere with the primary purpose of its use, which is the social interaction with other people in the environment. Thus, it becomes important to provide these robots with some autonomy, so that they are able to take some simple decisions during navigation when needed. This is crucial especially in unknown and dynamic environments where the robot will interact with people, objects and even other robots.

Note as well that this is a concern mostly only for *telepresence* robots. Indeed, for rescue robots, for example, the primary task is operating the robot. Thus, the operator is typically dedicating 100% of attention to the robot, which is appropriate for the application. Not so for telepresence robots, where often conversation and personal interaction are the primary tasks, and robot operation just a subsidiary need.

Thus there is a need to further reduce the cognitive load required to teleoperate a robot, beyond that provided by existing collision avoidance systems. In this article, we bring assisted driving to a new level, by introducing concepts

This work was supported by Microsoft Research.

Douglas G. Macharet is with the Computer Vision and Robotic Laboratory (VeRLab), Computer Science Department, Universidade Federal de Minas Gerais, MG, Brazil `doug@dcc.ufmg.br`

Dinei A. Florencio is with Microsoft Research, One Microsoft Way, Redmond, WA, USA `dinei@microsoft.com`

from collaborative driving to a telepresence robot. As before, the main objectives are to facilitate the control tasks performed by the operator, and to ensure a safe navigation for both the robot and the people around. However, instead of interpreting commands as direct motion requests, we use the input from the operator only as a general guidance to the target direction. This idea of a varying degree of autonomy is crucial for collaborative driving, usually applied to automobile automated driving in highways [7], [8]. In this paper, we use similar concepts, coupling the driver's guidance with a variable degree of autonomy to the robot, depending on the task and the environment. Previous work on telepresence [9] (and other teleoperated [10], [11]) robots has shown collision avoidance makes operation easier and reduce the number of collisions. In addition, (and in contrast to traditional collision avoidance systems) our approach also reduces the time required to complete a circuit, making navigation easier, safer, and faster.

The rest of the paper is organized as follows: in section II we summarize related work. In section III we introduce the proposed methodology, including descriptions of the semi-autonomous, autonomous modes, and of some elements of the (preliminary) user interface. Section IV presents some of the results, including a controlled user study (N=18). Section V present some conclusion and future work.

II. RELATED WORK

Several recent studies confirm the benefits of using telepresence robots in work and other social environments. Our robot most closely follows the concept of a Embodied Social Proxy (ESP) as presented in [12]. The ESP has the objective of being a physical representation of a remotely located member of a team. The study shows that the interaction between the remote member and the rest of team improved by the use of the ESP (in both formal and informal situations). However, although the ESP is mounted on a mobile platform, it has to be moved with the help of another person, since the platform has no actuators.

Teleoperated robots are already used in many areas and applications, however its use as a form of telepresence is still indeed recent, and brings new issues which are relevant to research. Use cases of telepresence robots in the workplace of different companies are presented in [13], [14], [15], [16]. In [13] the author shares his own experience of using a robot during one week in his office, which he concludes can bring benefits, especially to the remote teammate. In [14], [15], [16] different situations were analyzed considering both the people interacting through the robots and the people interacting with robots. They've shown the advantages you have when you use this technology, especially in less formal situations, such as a hallway conversation. But they also point the challenges and advances that still must occur in order to allow expansion in the use of telepresence robots.

Focusing on this difficulties, some guidelines for the the development of telepresence robots are presented in [17], [18]. Based on previous analysis and observations, these papers suggest some key features that a robot should have in order to improve both the remote person and the local team experiences, such as a good user interface and an assistive layer for the robot navigation.

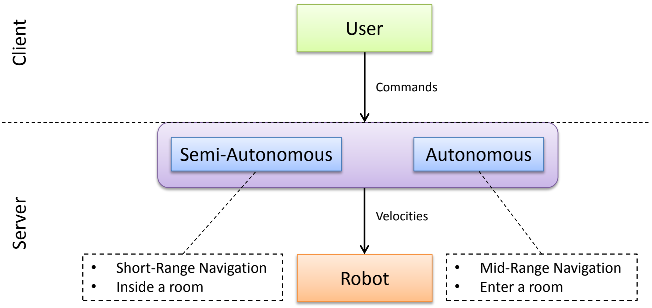


Fig. 1. Diagram of the system.

In [9], a simple assisted teleoperation architecture is presented. Based on a costmap generated using a laser range finder the robot can avoid obstacles and determine if it is safe to go through a certain path. Controlled user evaluation is realized showing that with the help of the assisted driving the number of collisions was reduced, but the time to complete the circuit was increased.

Collision avoidance has been investigated in a number of research papers [10], [9], [19], [20], [21]. In this paper we build on those technologies, by focusing not only on the obstacles, but also in the *empty* spaces. For example, assume the user seems to be driving the robot into a wall. Instead of slowing down the robot to avoid the collision, we'll investigate the *free space* in that vicinity. We may then, for example, infer the intended target is a nearby door.

Others studies (e.g. [22], [23]) show that not just the business environment can benefit from telepresence robots, but also people with special needs, such as the elderly and people with disabilities.

III. METHODOLOGY

The assistive framework proposed here focuses on two different scenarios, which we refer as short-range navigation and mid-range navigation. Fig. 1 presents and overview of the methodology.

The user sends high-level commands to the robot using the user interface presented in the remote computer. Each scenario has its own set of commands, and uses separate computations. Depending on the command issued, the corresponding module is used to compute the velocities that will be applied to the robot. Details of each scenario are presented in the next sections.

A. Short-range Navigation (Semi-Autonomous)

Often, operators do not want to be just spectators during the robot's navigation; they want to *control* it themselves. Thus, in the semi-autonomous mode, the objective is to share the control with the robot. More specifically, the user can issue low-level commands, which will be then transformed to provide low-level *safe* commands. Thus, it provides a certain degree of autonomy while the safety layer provides confidence and assurance to the user. The semi-autonomous navigation has applications is short-range navigation such as moving within a small environment (e.g a meeting room) without colliding with people or other objects.

For this scenario, our methodology is based on Potential Fields (PF)[19]. This technique treats the robot as a point under the influence of an artificial potential field $U(q)$, where $q = (x, y)$ is the position of the robot. We further decompose $U(q)$ into a repulsive field $U_r(q)$ (e.g., accounting for obstacles) and an attractive field $U_a(q)$ (e.g., accounting for the desired movement). Assuming differentiable potential fields, the force acting on the robot can be computed as:

$$\begin{aligned} F(q) &= -\nabla U(q) \\ &= -\nabla U_a(q) - \nabla U_r(q) \\ &= F_a(q) + F_r(q). \end{aligned} \tag{1}$$

The main advantages of this technique are its low computational cost and that — being a reactive technique — it can easily deal with a dynamic and unknown environment.

Unlike the classical PF technique, when the user is manually controlling the robot we do not have a defined goal acting as the attractive force field. Therefore, we consider that the user is directly controlling the components of the attractive field.

To make the framework as accessible as possible, we assume the user will be using just a keyboard (arrows keys). However, differently from other user interfaces (e.g. joystick), a keyboard has sharp variations (a key is either down or up). If mapped directly to the robot velocities this would produce abrupt changes in speed and/or directions making it even harder to drive. To solve this problem of abrupt accelerations we propose a smooth variation of the attractive force field, given by:

$$U_a(q) = U_a^{\min} + \Delta u_a. \tag{2}$$

When the user sends a command, we start with a minimum field $U_a^{\min}$ and gradually increment the magnitude of the field by a value Δu_a every second (or any other desired Δt), as long as the user is still sending commands. The value of Δu_a can be adjusted by the user before starting the navigation, and is expected that the user may increase it as he feels more familiar with the operation. Note also that we apply this smooth acceleration only to the linear velocity (i.e., angular velocities are constant).

Therefore, it is possible to navigate in small spaces at low speeds, but also achieve higher speeds when necessary (e.g. traveling over a long distance in a corridor), and most importantly, the control is transparent to the user (no need to worry about the speed).

The repulsive potential has influence when the robot gets closer than a certain range ρ_0 to an obstacle. Its magnitude is obtained by:

$$U_r(q) = \begin{cases} \frac{k_r}{2}\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)^2, & \text{if } \rho(q) \leq \rho_0 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where k_r is a scaling factor and $\rho(q)$ is the distance to an obstacle.

We receive as input a 2D laser read and a 3D point cloud from the Kinect camera. It is important to use a three-dimensional sensor, since the laser can only detect obstacles that intersect its plane. Thus, we project the 3D data in the floor plane and a raytracing is executed from the origin to the maximum range of the reading, varying in the interval $[0^\circ, 180^\circ]$. The result is a simulated 2D of all the closest obstacles in the space.

After the forces acting over the robot in q are calculated, we apply the controller given by Equation 4 to obtain the linear (v) and angular (ω) velocities values,

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} k_v F_x(q) \\ k_\omega F_y(q) \end{bmatrix} \tag{4}$$

where k_v and k_ω are scaling factors.

Another important issue within this scenario is communication loss between the operator and the robot, which may be very common in wireless environments [6]. Most telepresence systems today simply stop the robot if it loses connection with the wireless network. However, in the case that the operator loses connection, the robot may still continue running the last command sent, which is not the desired behavior.

We use a timeout in the PF controller in order to solve this problem. If the robot is still moving (have not received a stop command due to some network problem), but the time elapsed since the last command is over a certain time interval Δt_o (we used $\Delta t_o = 600$ ms) the robot stops for safety.

B. Mid-range Navigation (Autonomous)

There are some cases in which the user does not want to explicitly control the robot, such as tedious or exhausting tasks (e.g. go to the end of a corridor or go through a narrow passage). In such scenarios, the system allows the operator to adjust the autonomy of the robot, giving it full control on the navigation for a short term.

The first step consists of providing the user with the possible free directions that would be desirable to increase the robot's autonomy. We propose here the use of a discrete polar occupancy map $\mathcal{P}$ of the environment, inspired by works such as [20], [21].

The space around the robot is divided in n different angular sectors $\mathcal{P}_i$ of similar (angular) size. Each sector is centered in the origin of the robot's reference system and its value is computed as:

$$\mathcal{P}_i = \begin{cases} 1, & \text{if } \delta_i(L) < \mu \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where L is the list of obstacle points measured and $\delta_i(L)$ is a function that returns the minimum distance to an obstacle in sector i. Then, sector $\mathcal{P}_i$ is assumed to be occupied if an obstacle is closer then a certain threshold μ.

The next step is to obtain the free directions. In order to accomplish this we first identify the empty sectors ($\mathcal{P}_i = 0$).

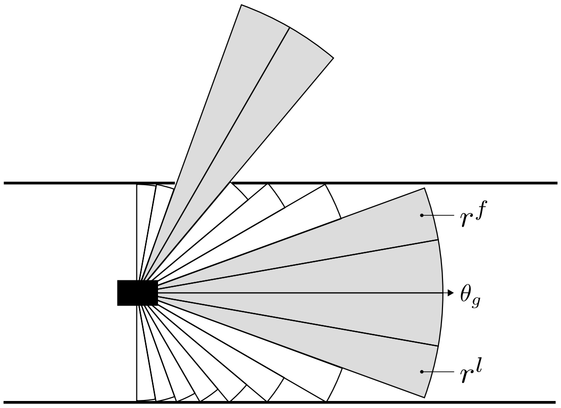


Fig. 2. Empty sectors around the robot and suggested goal direction.

Contiguous empty sectors are grouped to form a region r. The possible goal directions are then obtained accordingly to Equation 6,

$$\mathcal{F} = \left\{ \theta_g = \frac{r^f + r^l}{2} : \forall r \in \mathcal{R}, \tau(r) \geq s_{\min} \right\} \quad (6)$$

where $\mathcal{R}$ is the list of regions (r^f and r^l are respectively the first and last sector of a region), τ is a function that returns the size of a region (number of sectors) and $s_{\min}$ is the minimum region size to consider it a free possible direction.

Fig. 2 illustrates the complete process. The black box represents a robot in the middle of a corridor, empty sectors are depicted in gray. An option of navigation is then presented to the user pointing to the middle of the region. Only the region in front of the robot satisfies the minimum region size restriction (in the example $s_{\min} = 4$).

It is important to choose a proper value for $s_{\min}$. If the value is too large it can be difficult to find possible directions in very cluttered places, in the other hand, if the value is too low the system will identify free directions that are not important to the user.

After a direction is selected by the user, a goal position is defined in the neighborhood of the robot. Based on the robot's coordinate system, the goal position is calculated as:

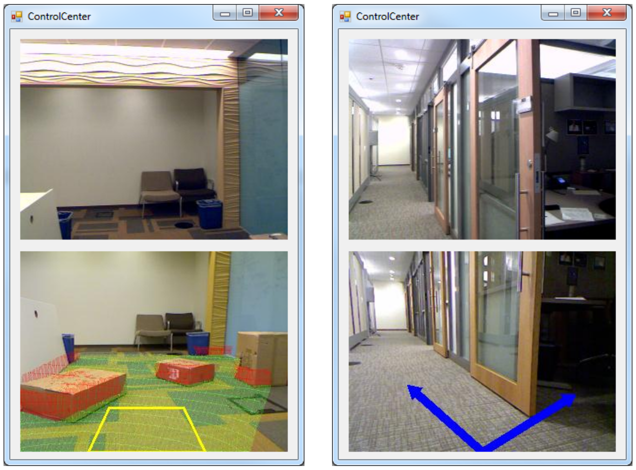
$$\begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} \cos(\theta_g)\mu \\ \sin(\theta_g)\mu \end{bmatrix}. \quad (7)$$

The classical Vector Field Histogram (VFH) algorithm [20] is then used to navigate the robot from its actual position to the given goal position autonomously (the user is able to retake control at any moment).

C. User Interface

The user interface is an important component in teleoperated systems and demands attention [24], [22], [18], since its the only connection between the operator and the environment where the robot is placed.

Based on the previous mentioned studies, we decided to design an interface as simple as possible in order not to overwhelm the user with unnecessary information.



(a) (b)

Fig. 3. User Interface used by the operator to control the robot.

The operator can see the environment through two different camera views. Both cameras point to the same direction and are separated by a certain height. The top camera is mainly used to interact with people, while the bottom camera is used to help navigate the robot.

The user can enhance the bottom camera view with two types of extra information. Although the user can request these at any time, each one roughly correspond to one of the navigation modes. The first one is an obstacle map (Fig. 3(a)), where free regions are green and possible obstacles are red. Yellow guide-lines (similar to those used in car parking systems) are also presented, making it easier to the user to identify the boundaries of the robot. The second option for extra information is the possible goal directions (obtained from the autonomous navigation module), presented to the user on the bottom view (blue arrows on Fig. 3(b)).The user can selected a direction by clicking on the related arrow (using the mouse).

IV. EXPERIMENTS

The telepresence robot used in the experiments was designed and built in-house. The platform is composed by a differential drive system, a low-cost laser range finder, a webcam, a Microsoft Kinect [25] and a speakerphone. The framework was developed using the Microsoft Robotics Developer Studio [26]. Figure 4 shows the robot and the arrangement of the sensors.

A. User Studies

The experiments were conducted with $N = 18$ participants (9 men and 9 women). Before the experiment, participants completed a questionnaire informing age, gender, and rating their prior experience playing video-games and using a robotic platform. None of these showed statically significant correlation with the results.

1) Experiment 1: The first experiment was designed to evaluate the collaborative layer. Remember this layer was developed to make navigation safer and easier in comparison to the control with no assistance.

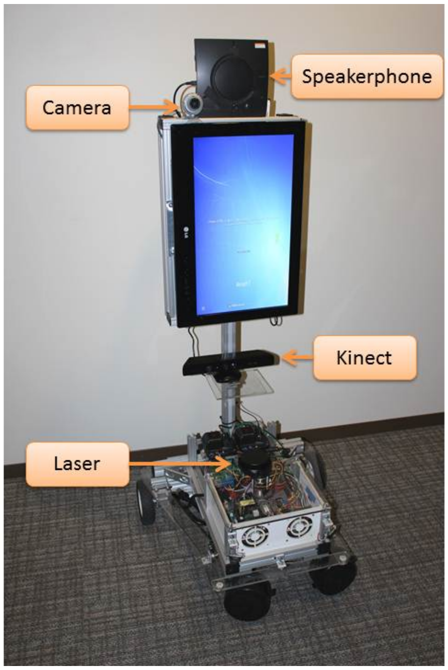


Fig. 4. Telepresence robot used in experiments.

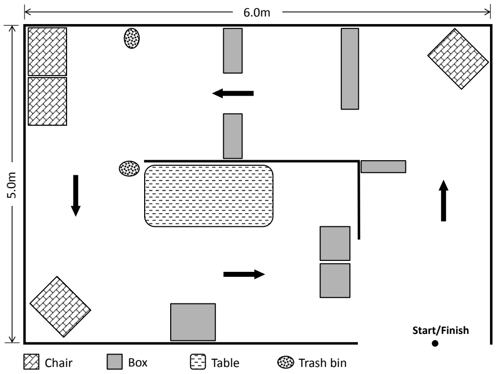


Fig. 5. Obstacle course used for the collaborative control system evaluation.

None of the participants had prior experience driving the telepresence robot, and they had a training session before the experiment. The experiment started with a training session, consisting on instructions on how to drive the robot (forward, backward and make turns) and practice time, where they were free to drive the robot on a different place from where the experiment would be conducted. After the practice time the robot was re-positioned in order to start the experiment.

A map of the obstacle course used in the experiment is presented in Fig. 5. The participants should start and finish at the marked location and navigate through the course in a counterclockwise direction. Light objects such as boxes and trash bins were used in order not no damage the robot in case of a collision, but the course also had other objects commonly found in an office such as chairs and a table.

The participants were told to drive the robot through the obstacle course considering speed and precision to complete the task. The experimenter noted the number of collisions with obstacles and the time to complete the course. After the experiment the participants completed a questionnaire rating the difficulty of the task on a scale of 1 (very easy) to 5 (very difficult).

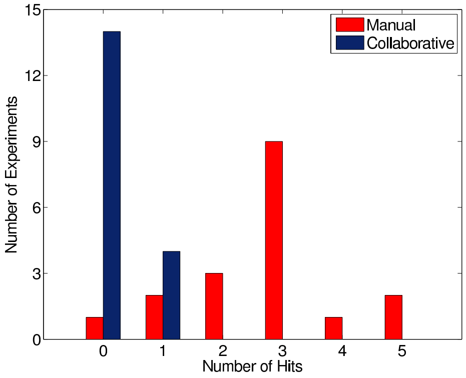


Fig. 6. Histogram of the the number of collisions during the experiments.

The participants repeated the course using a different control type on the second time. The control type (collaborative control or fully manual control) used on the first run was randomly chosen, where 5 of the men and 5 of the women started with the fully manual control (no obstacle avoidance) and the others participants started with the collaborative control. They were asked to rate the difficulty again after the second run.

Participants were told when the collaborative was being used and how it would help avoid collisions by moving the robot away from the obstacles.

It is important to mention that in the case of a collision, the experimenter would remove that obstacle from the way if necessary. This was done as time to overcome an obstacle after a collision was often significantly high. By removing the obstacle, we are trying to ensure that the actual time to complete the course is being measured (still counting the number of collisions).

A comparison of the average number of collisions and time to complete the circuit using the different control types is presented in Table I.

TABLE I

PERFORMANCE COMPARISON BETWEEN MANUAL VS. COLLABORATIVE CONTROL - MEAN (STANDARD DEVIATION).

Type of Control	Number of Collisions	Time (s)
Manual	2.72 (1.27)	132.9 (34.3)
Collaborative	0.22 (0.42)	113.0 (39.2)

The collaborative control helped the operators complete the course with much fewer collisions (none in most cases), a t-test analysis give us $t(17) = 8.84$ and two-tail $p \approx 0$. These collisions happened with obstacles almost behind the robot during very sharp turns. We believe that making a better tuning of k_r would solve this problem.

A histogram of the number of collisions is presented in Fig. 6, allowing a better overview of the results for all experiments.

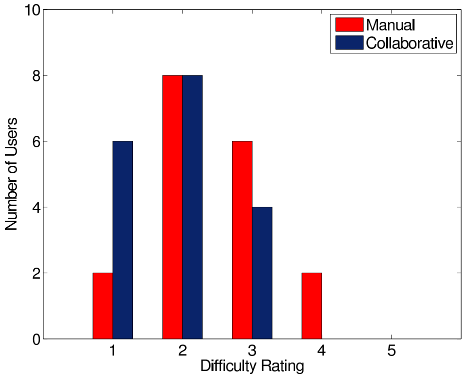


Fig. 7. Difficulty rating of the task using the different types of control.

The proposed methodology also reduced the time to complete the task. The mean time reduction ($\mu = 19.9$, $\sigma = 17.1$) was significantly greater than zero, $t(17) = 4.94$, two-tail $p = 0.0001$. The 95% confidence interval (CI) for mean time reduction is (11.4, 28.4). This result outperforms the one obtained in [9] (where the technique actually increased the time to complete the task). The main reason is that the proposed methodology will not only avoid the obstacles but can also help the operator during the navigation (e.g. making a turn or going though a narrow passage).

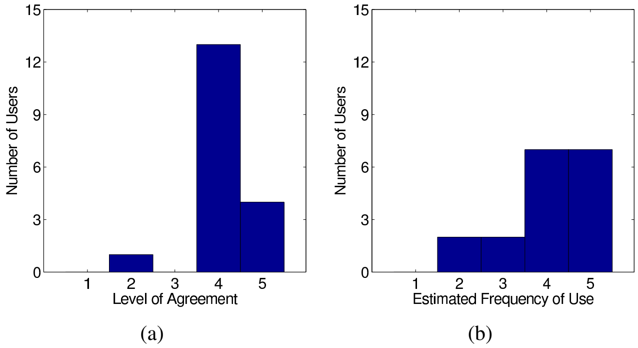
The participants that took longer to finish the course using the collaborative control pointed as the main reason the lack of visual feedback when the system was autonomously avoiding the obstacles (the system can stop the robot if the operator tries to run over an obstacle at certain frontal angles).

As it can be seen in Fig. 7, most operators considered the task as easy in both manual control ($\mu = 2.44$ and $\sigma = 0.85$) and collaborative control ($\mu = 1.89$ and $\sigma = 0.76$). Furthermore, driving using the collaborative control was considered slightly easier ($t(17) = 3.34$ and two-tail $p = 0.003$).

The lack of visual feedback was pointed again as the reason to rate the collaborative control as more difficult than the manual control. Participants mentioned that when using the collaborative control the robot wasn't behaving exactly as the commands they were sending (this happens when the user is in a collision route and the robot move away from the obstacle).

After the second run, the participants could write general comments about the system. The most common complaint of the participants was the narrow field of view of the bottom camera, which made it difficult to visualize and avoid obstacles in the environment.

A few participants also mentioned that it was difficult to drive the robot without knowing the size of the robot in relation to the image being displayed (bottom camera view). It is important to mention that the enhanced information feature was disabled during the experiments and users didn't know about it. These comments show how important this information is and why we chose to display it on the user interface.



(a)

(b)

Fig. 8. Experiment 2 analysis.

2) Experiment 2: The second experiment measured the perception of the participants in relation to the possible directions pointed by the system. We have also estimated the usefulness of the feature by asking users to estimate their frequency of usage.

The robot was placed in a corridor where the system would suggest different possible directions such as go along the corridor, enter a room or go into an open space. Participants were requested to drive the robot straight forward through the corridor, make a turn and go back to the original position. During the navigation they were asked to pay close attention to the proposed directions being presented.

After completing the designated task, participants were asked to rate their level of agreement with the directions presented during the experiment on a scale of 1 (totally disagree) to 5 (totally agree). The participants were also asked to estimate how often they would use this feature (navigate the robot by selecting a direction) on a scale of 1 (never) to 5 (always).

A histogram presenting the overall rating for the level of agreement is presented in Fig. 8(a). Most users had a high level of agreement with the directions showed ($\mu = 4.11$, $\sigma = 0.68$). The 95% CI for this rating was (3.77, 4.45).

Fig. 8(b) shows the rating for the estimated frequency of use of this feature. Most of the participants commented that they would preferably use the "click on the direction" control over using the keyboard ($\mu = 4.06$, $\sigma = 1$). The 95% CI for this rating was (3.56, 4.55).

The field of view was criticized once again, since the system can point to a direction that is off the field of view (e.g. the entrance of a room). Participants mentioned that it is not intuitive to point to a direction they cannot see at the moment.

V. CONCLUSIONS AND FUTURE WORK

This work presented a collaborative control framework to facilitate the control of telepresence robots by remote human operators. Instead of focusing only on avoiding *obstacles* (as previous collision-avoidance systems), the proposed framework smoothly integrates information about *free spaces*. This allows to re-interpret user commands at a higher level, facilitating tasks as going through narrow passages or doors, or even simple turn making.

The results are promising: users had significantly fewer hits (none in most cases) and took less time to complete the task using the proposed methodology. Most users also agreed with the directions of interest showed by the system and confirmed that they would probably use it very often (more than the keyboard).

The projects still needs improvement in a number of fronts. An obvious next step is to develop the large-range navigation. In this case, the user will be able to use a map and select an objective for the robot (e.g go to a specific room). This will reduce even more the need of human supervision in time consuming tasks. We left this for later work, as research in autonomous navigation (while still not perfect) is a reasonably mature field.

Regarding the design of the robot, one of the clear outcomes of the user studies was the desire for better visualization of the environment. Thus, we are planning to add a wide-angle lens in replacement of the bottom camera. Two additional Kinect sensors will also be added to the robot. These will help improve the obstacle avoidance layer (making it possible to detect 3D obstacle in a larger neighborhood around the robot). Additionally, a wider coverage will also help in the suggested directions, which is based on available path. Finally, with a Kinect with smaller minimum range now available, we will reposition (tilt lower) the two side Kinects to provide better coverage of nearby obstacles.

The user interface also needs improvement. In particular, visual (and possibly auditory) feedback will be presented to the user when the robot is behaving autonomously (avoiding obstacles). We are also planning to port the user interface making it accessible through the web. This will make it easier to the user to use system remotely, being able to control the robot even using mobile devices such as smartphones.

Although not mentioned in this paper, we are also investigating other aspects directly related to the telepresence use of the robot, including speech quality and noise reduction [6], [27], [28], [29].

Future research directions also include the study and addition of social accepted behaviors on the robot navigation. People are now treated as simple obstacles. In the future, we plan to create a special class for them, adding dynamic and social considerations.

VI. ACKNOWLEDGMENTS

The authors would like to thank Bruce Cleary and Tom Blank for the support on the construction and integration of the mechanical system used in the experiments.

REFERENCES

[1] "Anybots - remote yourself," http://www.anybots.com, [Online; accessed 15-January-2012].

[2] "VGo robotic telepresence for healthcare, education and business," http://www.vgocom.com, [Online; accessed 16-January-2012].

[3] "Suitable technologies - remote presence, social presence, telepresence solutions," http://www.suitabletech.com, [Online; accessed 17-January-2012].

[4] "MantaroBot telepresence robots," http://www.mantarobot.com, [Online; accessed 17-January-2012].

[5] "iRobot corporation: Home page," http://www.irobot.com, [Online; accessed 12-January-2012].

[6] A. Conceicao, J. Li, D. Florencio, and F. Kon, "Is IEEE 802.11 ready for VoIP?" in *IEEE Int. Workshop on Multimedia Signal Proc. (MMSP)*, 2006.

[7] J. Wei and J. Dolan, "A multi-level collaborative driving framework for autonomous vehicles," in *18th IEEE Int. Symposium on Robot and Human Interactive Communication (Ro-Man)*, 2009.

[8] N. Gaubert, M. Beauregard, F. Michaud, and J. de Lafontaine, "Emulation of collaborative driving systems using mobile robots," in *IEEE Int. Conf. on Systems, Man and Cybernetics*, 2003.

[9] L. Takayama, E. Marder-Eppstein, H. Harris, and J. M. Beer, "Assisted driving of a mobile remote presence system: System design and controlled user evaluation," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[10] H. Yuste, L. Armesto, and J. Tornero, "Collision avoidance systems for manual-assisted driving," *41st Int. Symposium on Robotics (ISR)*, 2010.

[11] Q. Zeng, E. Burdet, B. Rebsamen, and C. L. Teo, "Evaluation of the collaborative wheelchair assistant system," in *IEEE 10th Int. Conf. on Rehabilitation Robotics (ICORR)*, 2007.

[12] G. Venolia, J. Tang, R. Cervantes, S. Bly, G. Robertson, B. Lee, and K. Inkpen, "Embodied social proxy: mediating interpersonal connection in hub-and-satellite teams," in *28th Int. Conf. on Human Factors in Computing Systems (CHI)*, 2010.

[13] E. Guizzo, "When my avatar went to work," *IEEE Spectrum*, vol. 47, no. 9, pp. 26–50, september 2010.

[14] M. K. Lee and L. Takayama, "'now, i have a body': uses and social norms for mobile remote presence in the workplace," in *29th Int. Conf. on Human Factors in Computing Systems (CHI)*, 2011.

[15] K. M. Tsui, M. Desai, H. A. Yanco, and C. Uhlik, "Exploring use cases for telepresence robots," in *6th Int. Conf. on Human-Robot Interaction (HRI)*, 2011.

[16] K. M. Tsui, M. Desai, and H. A. Yanco, "Towards measuring the quality of interaction: Communication through telepresence robots," in *Performance Metrics for Intelligent Systems Workshop (PerMIS)*, 2012.

[17] K. M. Tsui, M. Desai, H. A. Yanco, and C. Uhlik, "Telepresence robots roam the halls of my office building," in *Human-Robot Interaction (HRI) Workshop on Social Robotic Telepresence*, 2011.

[18] M. Desai, K. Tsui, H. Yanco, and C. Uhlik, "Essential features of telepresence robots," in *IEEE Conf. on Technologies for Practical Robot Applications (TePRA'11)*, 2011.

[19] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. Scituate, MA, USA: Bradford Company, 2004.

[20] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, jun 1991.

[21] J. Minguez and L. Montano, "Nearness diagram (ND) navigation: collision avoidance in troublesome scenarios," *IEEE Trans. on Robotics and Automation*, vol. 20, no. 1, pp. 45–59, feb. 2004.

[22] K. M. Tsui, A. Norton, D. Brooks, H. A. Yanco, and D. Kontak, "Designing telepresence robot systems for use by people with special needs," in *Int. Symposium on Quality of Life Technologies: Intelligent Systems for Better Living*, 2011.

[23] J. M. Beer and L. Takayama, "Mobile remote presence systems for older adults: acceptance, benefits, and concerns," in *6th Int. Conf. on Human-Robot Interaction (HRI)*, 2011.

[24] B. Keyes, M. Micire, J. L. Drury, and H. A. Yanco., *mproving Human-Robot Interaction through Interface Evolution*. Human-Robot Interaction, Daisuke Chugo (Ed.) InTech, 2010.

[25] Kinect, "Kinect - Xbox.com." http://www.xbox.com/kinect, [Online; Accessed: 21-July-2011].

[26] MRDS, "Microsoft Robotics Developer Studio." http://www.microsoft.com/robotics, [Online; Accessed: 01-July-2011].

[27] Y. Rui, D. Florencio, W. Lam, and J. Su, "Sound source localization for circular arrays of directional microphones," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005.

[28] D. Florencio and H. Malvar, "Multichannel filtering for optimum noise reduction in microphone arrays," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001.

[29] F. Ribeiro, C. Zhang, D. Florencio, and D. Ba, "Using reverberation to improve range and elevation discrimination for small array sound source localization," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 7, pp. 1781 –1792, sept. 2010.