Egocentric and Exocentric Teleoperation Interface using Real-time, 3D Video Projection

François Ferland
Université de Sherbrooke
Sherbrooke, Québec, Canada
francois.ferland@
usherbrooke.ca

François Pomerleau
Université de Sherbrooke
Sherbrooke, Québec, Canada
francois.pomerleau@
usherbrooke.ca

Chon Tam Le Dinh
Université de Sherbrooke
Sherbrooke, Québec, Canada
chon.tam.le.dinh@
usherbrooke.ca

François Michaud
Université de Sherbrooke
Sherbrooke, Québec, Canada
francois.michaud@
usherbrooke.ca

ABSTRACT

The user interface is the central element of a telepresence robotic system and its visualization modalities greatly affect the operator's situation awareness, and thus its performance. Depending on the task at hand and the operator's preferences, going from ego- and exocentric viewpoints and improving the depth representation can provide better perspectives of the operation environment. Our system, which combines a 3D reconstruction of the environment using laser range finder readings with two video projection methods, allows the operator to easily switch from ego- to exocentric viewpoints. This paper presents the interface developed and demonstrates its capabilities by having 13 operators teleoperate a mobile robot in a navigation task.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Graphical user interfaces, Input device and strategies (e.g., mouse, touchscreen), Screen design (e.g., text, graphics, color)

General Terms

Algorithms, Design, Experimentation, Human Factors, Measurement

Keywords

Enabling technologies, Feedback modalities, Interface design and usability, Human-Robot Interaction (HRI), Robot intermediaries (e.g., telepresence, proxies, avatars), Situation Awareness (SA)

1. INTRODUCTION

The user interface is the central element of a telepresence robotic system. It is through this interface that users can interact remotely with the intended environment, via the robot. Such interfaces must be designed in accordance with user's skills and application purposes, and not from the designer's standpoint, which is too often the case [1]. Information display, layout and visualization capabilities directly affect teleoperation performance. For instance, displaying the robot's reference frame and status, combining readings from multiple sensors, providing the ability to inspect it and automatically presenting contextually-appropriate information (e.g., showing a rear-facing camera when moving backward) all need to be considered in teleoperation interfaces for urban search and rescue (USAR) applications [16].

Because it provides a general and multi-dimensional (2 dimensional with color, texture) view of the remote environment, a video display is found in most interfaces. It can however be enhanced by adding other information (e.g., laser range finder, virtual reality) to improve situation awareness [20, 6] while minimizing the operator's cognitive load [22]. Such fusion must be done in appropriate ways, and simply using distinct windows results in operators focusing mostly on the video display [2, 5, 19]. Combining information modalities is also influenced by the visualization viewpoint, which can be egocentric (seeing the world from the robot's perspective) or exocentric (observing the world from an external perspective). An egocentric viewpoint is usually better for navigating and avoiding obstacles, while an exocentric viewpoint gives the operator a better understanding of the environment's structure [15]. Egocentric multi-modal displays can superimpose graphical information to a 2D video feed (e.g., distance markers and translucent 2D maps [21], infrared camera superposition for heat sources detection [7]) or superimpose a 2D video feed with a virtual reality model of the world [9]. Proportion and localization issues are then key challenges to address. In interfaces following the ecological paradigm [12], where the teleoperated robot is displayed along with its environment, exocentric multi-modal displays can be made by constructing 3D structures from exclusively

HRI'09, March 11–13, 2009, San Diego, California, USA.

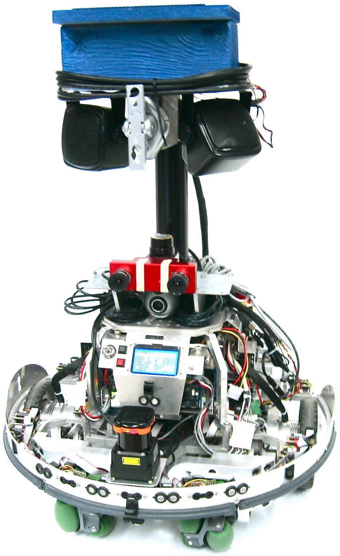


Figure 1: *Telerobot*, a teleoperated robot for assistive homecare.

2D data [15]. However, displaying video on a fixed 2D plane in front of the robot introduces a serious occlusion problem, because the 2D plane can completely hide the 3D extrusion from the range finder readings. Finally, including the representation of the robot's chassis in either ego- or exocentric views contributes significantly to situation awareness [8].

As a logical extension to such observations, this paper demonstrates that it is possible to use 3D data representations in real-time in a new interface that 1) combines laser range finder data with 3D video projections from a stereo camera (to handle occlusion problems), and 2) provides the operator with the ability to change viewpoints as desired. After presenting the hardware and software components used in our experimental settings to situate the capabilities and constraints of our development, the paper describes our ego-/exocentric 3D interface and the two video projection methods implemented. It also reports observations made from having 13 operators use the interface in a navigation task in terms of operator's gaze, viewpoints, control and task accuracy, situation awareness and usefulness. This pilot study was meant to validate the possibility of evaluating precise metrics of our system in preparation of a performance-based comparison of different display modalities.

2. EXPERIMENTAL PLATFORM

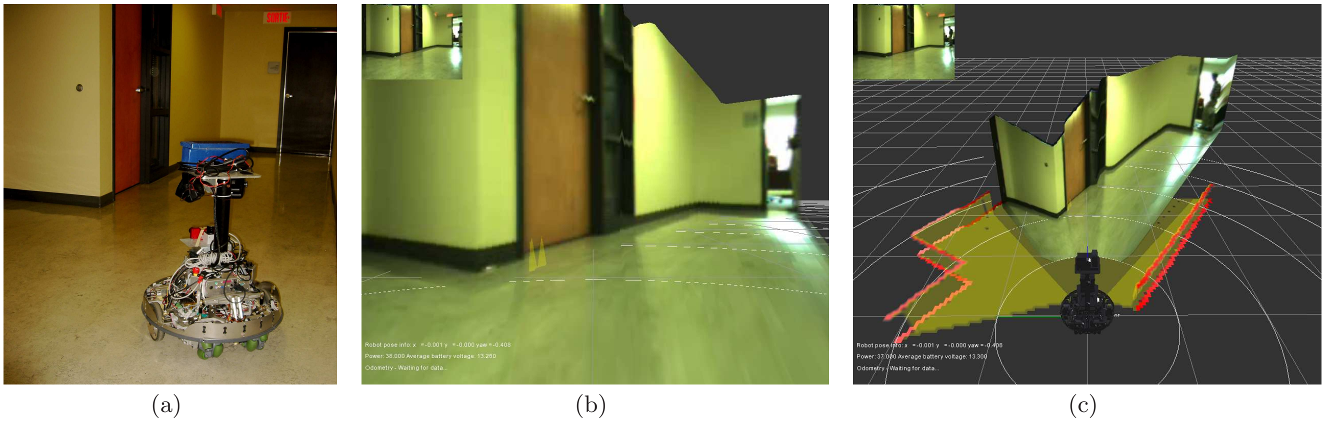
Our experimental setup makes use of Telerobot [9, 10], a differential drive robot platform shown in Figure 1 and designed to provide health care assistance in home environments. The robot's onboard computer is an Intel Pentium® M processor at 1.7 GHz, 512 MB of RAM and using a 802.11B wireless LAN card. The laser range finder is a Hokuyo URG-04LX placed approximately at 0.4 m from the ground. This particular model has a 270 degrees field of view over a range of 4 meters. A Videre Design STOC (*STereo On a Chip*) stereo camera (with two (320×240) video feeds) is placed between a webcam (used for other purposes) and the laser range finder. This position provides a good view of the ground in front of the platform and does not significantly hinder the recognition of higher-placed objects or individuals. This Videre camera model processes disparity information on a onboard FPGA, avoiding taking up processing time on the robot's onboard computer. A Logitech® wireless gamepad, which features two analog joysticks and ten digital buttons, is used to teleoperate the platform.

The teleoperation interface has been mostly developed and entirely tested on an laptop computer featuring an Intel Core® Duo clocked at 1.83GHz with 2GB of RAM and an ATI® X1600 GPU with 128 MB of dedicated video memory. The developed software is distributed on both the robot's onboard computer and the operator's laptop. The developed server reads from various sensors, calculates odometry, sends data at 10 Hz over the network, receives asynchronous command from the client's software and applies them to the motor drives. A set of libraries were developed to manage user inputs and communicate with robot servers. Except for stopping automatically the motors when the network connection is lost, there are no security modalities applied to the operator's commands. Video acquisition and transmission is done concurrently on a distinct TCP/IP connection. Once acquired on the robot, the left image and the disparity information are encoded in a single, double-width image (640 ×240) and transmitted at 10 Hz using the FFMPEG H.263+ codec implementation to compress data at 500 kbps. For the 3D display, we developed a simple library to manage the rendering of a asynchronously-updated scene graph. Our library requires OpenGL 2.1 and currently runs on Mac OS X and Linux. Asynchronous incoming data, represented as events, immediately updates internal structures. These changes are picked up by various synchronized threads, notably our rendering thread which needs to be closely coupled with the underlying operating system.

3. EGOCENTRIC AND EXOCENTRIC 3D INTERFACE

Figure 2 features screenshots of our ego-/exocentric 3D interface. The top-left window (parts (b) and (c)) shows the left video feed coming from the stereoscopic camera. The text on the bottom-left of the main window displays status information about the robot (e.g., the batteries' voltage level, odometry). The main window displays either the egocentric or the egocentric viewpoints, constructed using laser-based video projection. In the exocentric view, laser range finder data are extruded in red, and the yellow parts are zones that are considered empty. The displayed square grid has 1 m × 1 m cells and is fixed at the world's origin, and the circular grid, separated by 1 m, follows the robot.

Internal representation of the environment is based on a sparse 3D cell array with a granularity of 0.04 m. Incoming spatial data is continuously accumulated to provide the operator with a global map of the environment. For laser range finder data, our interface first maps the data in a 3D discretized point cloud, to then create a flattened 2D representation from the ground plane to create the local map. This local map's cells are then extruded at the same height as the laser range finder to create virtual, translucent walls. Odometry readings are used to position the robot and local maps. After having moved from more than either 0.25 m in translation or 10 degrees in rotation, local maps are com-



```
Robot pose info: x   = -0.001 y   = -0.000 yaw = -0.408
Power: 38.000 Average battery voltage: 13.250
Odometry - Waiting for data...
```

```
Robot pose info: x   = -0.001 y   = -0.000 yaw = -0.408
Power: 37.000 Average battery voltage: 13.300
Odometry - Waiting for data...
```

(a) (b) (c)

Figure 2: Screenshots from our ego-/exocentric 3D interface : (a) the scene; (b) egocentric viewpoint; (c) Exocentric viewpoint.

pared and registered using an Iterative Closest-Point (ICP) variant named Kd-ICP [14]. When the corrected position is obtained, the robot's model position in the interface is changed, a new local map is created and the 2D global map is updated. Because the dimensions of the global map are unknown at the initialization phase, the global map is managed as a set of sparsely located tiles of a fixed size. The global map resolution is independent of the original local maps and is defined as an amount of cells in a single square size covering 1 m in size. The tile set is indexed in a hash map. To place the incoming local map, our interface uses a single matrix translating and rotating its bitmap, built from the corrected position. This process, implemented using a set of OpenGL Shading Language (GLSL) vertex and fragment programs, provides an efficient way to display a large bitmap on a plane in 3D space. Global maps frequently reach 1280 ×1280 cells.

The robot's model is displayed using multi-element triangular meshes. Surface normals are defined for each triangle vertice. Each element can have a different material defined by its diffuse, specular and ambient color, and specular shininess factor. Rendering is done using a single directional light source with the Blinn-Phong [3] shading model. To create the robot's model, the complete SolidWorks® model is exported to a Virtual Reality Markup Language (VRML) file, rotated using Blender (an open-source application) to make it fit with our coordinate system and then exported to a Collada file, a XML-based format popular with 3D modeling animation packages. The Telerobot model features more than 400,000 triangles and does not tax significantly the system's rendering performance. Being able to provide a true-to-life representation of the operated robot is important for proportion representation in the 3D interface. Creating such models quickly and effortlessly facilitates the use of other robot platforms with our interface.

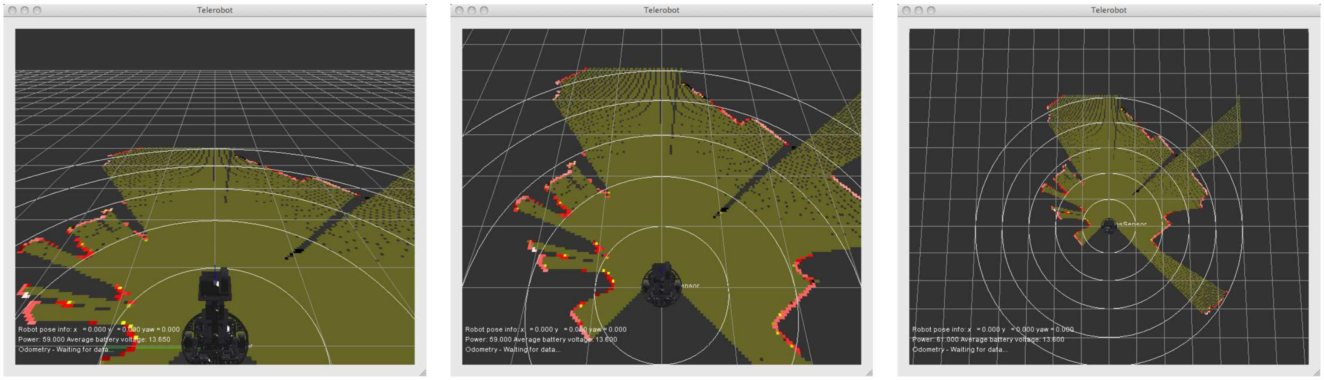
A virtual camera, acting as the operator's viewpoint and displaying the reconstructed 3D scene in the main window, is always following the robot by targeting a point located at 1 m over the ground and 1 m in front of it. This target point places the robot's model in the lower third portion the main window, with the virtual camera at its default position of 3 m ($\rho = 3$) behind the target, looking straight ahead ($\theta = 0$) at 30 degrees tilt angle from the ground ($\phi = \frac{\pi}{6}$). Figure 3 illustrates this configuration using a cylinder to represent the robot's base, an arrow for its heading and an eye for the operator's viewpoint. By modifying ϕ and keeping $\theta = 0$, the interface can switch from chase and track-up perspectives [4]. Adjustable perspectives have been noted as a way to reduce the operator's cognitive workload [12]. To enable this, the operator can continuously rotate and zoom around this target point using one of the analog joysticks and two buttons. In addition to these controls, the operator can cycle through five presets positions using a single button. These presets include a purely egocentric point of view and a top-down and track-up map-like one. Figure 4 shows screenshots of the three first exocentric presets, with the video projection and extrusion disabled.

According to Milgram's taxonomy of Mixed Reality (MR) of display integration [11], our system could be classified as variably-congruent. Its centricity is continuously variable, since we can move freely between ego- and exocentric viewpoints. Even though our control scheme is always ego-referenced, it closely matches the display's reference frame when the viewpoint is situated right behind our robot. However, the Control/Display (C/D) offset increases when the selected θ angle is farther from zero. Finally, our control order is 1 because our joystick is mapped to the speed of the robot.

To display the stereo camera data, our interface uses one of two video projection methods. The first simply projects color images from the video feed to the virtual representation of the world as set by laser range finder data, and the second projects stereoscopic data.

3.1 Laser-based Projection

The laser-based projection method creates a 3D mesh covering the floor and elevating virtual walls based on the laser range finder data. Disparity information from the stereoscopic camera is not considered. This mesh covers the field-of-view of the video feed and serves as a virtual projection screen. Each of the mesh's vertices is passed through conventional model, view and projection matrices describing the transformation from the mesh's geometry to 2D coordinates on the left video image plane. These coordinates represent texture coordinates used to map the video image on the mesh. As expressed by Equation 1, the vec-



Robot pose info: x = 0.000 y = 0.000 yaw = 0.000
Power: 59.000 Average battery voltage: 13.650
Odometry - Waiting for data...

(a) $\left(0, \frac{\pi}{6}, 3\right)$

Robot pose info: x = 0.000 y = 0.000 yaw = 0.000
Power: 59.000 Average battery voltage: 13.600
Odometry - Waiting for data...

(b) $\left(0, \frac{\pi}{3}, 4\right)$

Robot pose info: x = 0.000 y = 0.000 yaw = 0.000
Power: 61.000 Average battery voltage: 13.600
Odometry - Waiting for data...

(c) $\left(0, \frac{31\pi}{32}, 10\right)$

Figure 4: Screenshots of exocentric viewpoint presets, noted as (θ, ϕ, ρ) .

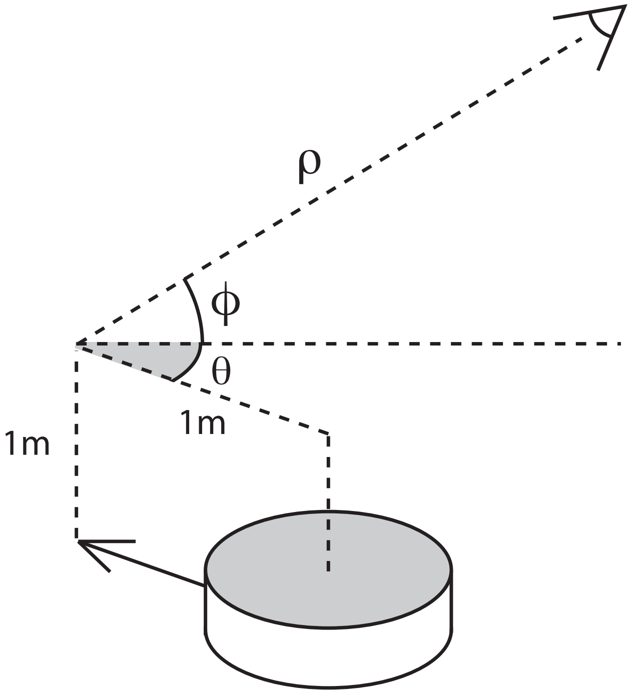


Figure 3: Viewpoint configuration.

tor $(mv_{14}, mv_{24}, mv_{34})^T$ represents the translation from the stereo camera's left viewpoint to the laser range finder's origin. In our case, the camera and the laser both have the same orientation, but a rotation matrix could be applied to the $ModelView$ matrix if the laser range finder and/or stereo camera were placed on a pan-tilt head, for instance.

$$ModelView = \begin{bmatrix} 1 & 0 & 0 & mv_{14} \\ 0 & 1 & 0 & mv_{24} \\ 0 & 0 & 1 & mv_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The projection matrix is expressed by Equation 2, where (t, b, l, r, n, f), for top, bottom, left, right, near and far, are the frustrum parameters, which can be calculated from the lens' horizontal field of view in radians (θ), the aspect ratio of the image (a) and a fixed near and far plane:

$$\begin{array}{rclrcl} a & = & w/h & & & \\ n & = & 1.0 & f & = & 1000.0 \\ w_h & = & n\sin(\theta/2) & h_h & = & \frac{w_h}{2} \\ l & = & -w_h & r & = & w_h \\ b & = & -h_h & t & = & h_h \end{array}$$

$$Projection = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad (2)$$

So, for a mesh vertex v, its location v' on the video image plane can be calculated using Equation 3.

$$v' = (v'_x, v'_y, v'_z, v'_w)^T = Projection \times (ModelView \times v) \quad (3)$$

After a conventional perspective division using v'_w, the normalized (from -1.0 to 1.0 in each dimension) 2D coordinate on the image frame v'' is derived from Equation 4:

$$v'' = \frac{(v'_x, v'_y)^T}{v'_w} \quad (4)$$

With the full set of v'', it is possible to apply standard texture mapping using OpenGL. In our setup, these texture coordinates are generated in a GLSL vertex shader used in a single rendering pass for the whole projection.

3.2 Stereoscopic Projection

This method exploits depth information that can be derived from the camera's stereoscopic disparity data. 2D image vectors (defined as $v_s = (x, y, d, 1)$) from such a camera can be projected in 3D space using the projection matrix defined by Equation 5, where a is the image's aspect ratio, f_s the lens' focal length and b_s the stereo camera baseline, i.e., the space between our head's optical axes.

$$StereoProjection = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & f_s \\ 0 & 0 & -\frac{1}{b_s} & 0 \end{bmatrix} \quad (5)$$

To efficiently transform entire images into a 3D representation at 10 Hz, we developed a set of GLSL fragment shaders to generate and render a resolution-independent triangular mesh that can be resized according to a system's processing abilities. Our first shader takes the single-channel disparity map as its input and applies the matrix of Equation 5 to vertices created from the normalized texture coordinates (between 0.0 and 1.0) for the x and y components and the sampled disparity at these same coordinates. The result is stored in a 4 channel, single precision float texture that is the size of the desired mesh. This texture is copied into a vertex buffer using a pixel buffer. The vertex buffer is then rendered using a constant index buffer connecting all coordinates into a square grid. The result is a deformed surface texture-mapped with the original left-camera image. We are currently using a 256×256 mesh, which is a little bit smaller than the image's resolution while still giving us acceptable results, since this mesh usually only covers a portion of the main window.

When sampling the disparity map, our method applies a local variance filter to reject pixels that are situated between zones that are far away from each other. By calculating the variance in a 3×3 window around each disparity sample, a mask of pixels going over a certain threshold is created. This mask acts as a second rendering target in the projection fragment shader and is used during the rendering stage to leave these fragments as transparent. This disparity threshold can be adjusted at runtime.

4. EXPERIMENTAL RESULTS

The objectives of the experiments conducted are to evaluate our ego-/exocentric 3D interface modalities in conditions that challenge both the operator's cognitive load and situation awareness when navigating in an unknown course and while accomplishing observational tasks. By letting the operator choose the visualization viewpoint, we also want to observe what preferences emerge for the tasks. The experimental scenario consists of teleoperating Telerobot in the area represented in Figure 5. The operators have not and cannot directly see the course and have two tasks to accomplish: 1) start at the black dot, locate table A, turn around it and count the number of visible orange cones; 2) follow a path identified by arrows on walls and furniture (at approximately 0.5 m from the ground). The course features different lightning conditions (grey: natural illumination; white: fluorescent lighting). A trial starts with an explanation of the system's control and display, followed with a short familiarization period. When ready, the operator starts executing the tasks. During a trial, at the locations identified by black triangles, the operator gets interrupted by the experimenter to estimate the distance between two colored tubes that are clearly visible from the video feed but not registered with the laser range finder. At the end of the course, the operators are instructed to go back to the starting position and place the robot at 2 m from the wall, directly aligned with a specific target placed at the height of the camera. One trial lasts from 15 and 45 minutes, and the maximum robot's velocity allowed is set to 0.3 m/s.

Using a faceLAB® system (developed by Seeing Machines Inc.), we measured the operator's eye gaze directed toward the interface and the reference sheet placed on the table below the screen. We also logged the position of the interface's virtual camera, collisions with obstacles, commands given to the robot and the time taken to accomplish the tasks. We also encouraged operators to think aloud during the experiment and tallied their comments using the LASSO technique [5]. Once the tasks accomplished, we asked operators to rank, from one to five, their impression of the usefulness of different features of our interface, i.e., the 2D Video Window (top-left window), Laser Mapping and Extrusion (red-yellow lines), the Adjustable Viewpoint feature, Laser-based Projection and Stereoscopic Projection.

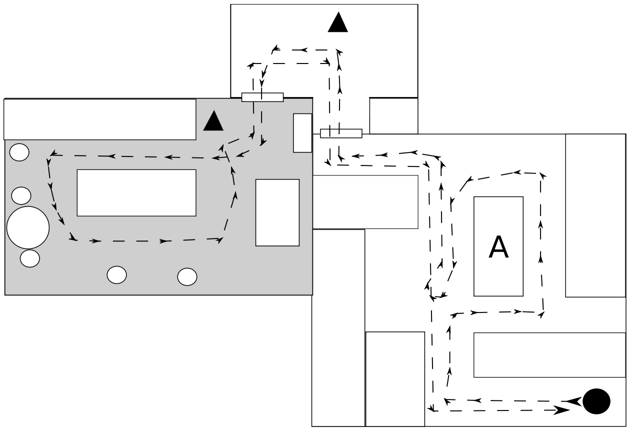


Figure 5: Experimental course.

The test population was made of a convenience sample of 13 voluntary male participants, aged from 19 to 58. Four had previous experience in teleoperating a mobile robot, seven mentioned playing videogames at least one hour per week, and two never worked on or used a robot of any form.

4.1 Gaze

Figure 6 represents the total distribution of the operators' gaze on the screen. A gaze point or glance represents a single sample recorded by the faceLAB® system running at 60 Hz. Glances at the 2D video window represent 4.8% of total trial samples. The gaze is mostly concentrated in the top two thirds of the screen, which corresponds, for exocentric viewpoints, to what lies ahead of the robot. Glances at our text display are extremely rare and coincide with the lack of utterances relative to the information there. The small size of the text might have played an important role in these results.

4.2 Viewpoint Positions

From the virtual camera's positions for all the trials (500 000 positions), we filtered out the initial default position, and each component of the viewpoint (θ, ϕ, ρ) were considered independently to produce the chosen viewpoint dataset, shown in Figure 7. Changes in θ, ϕ and ρ account for 35.9%, 68.7% and 27.2%, respectively. The θ distribution shows that the operators positioned the virtual viewpoint to look to the left ($\theta > 0$) more often than to the right, possibly because the tasks were mostly accomplished following a counter-clockwise path. The ϕ distribution suggests that a lower, straight-ahead tilt angle was preferred, especially for tasks requiring identifying objects in the scene, like directional arrows or orange cones. More vertical, top-down positions were selected for shorter period of time, usually

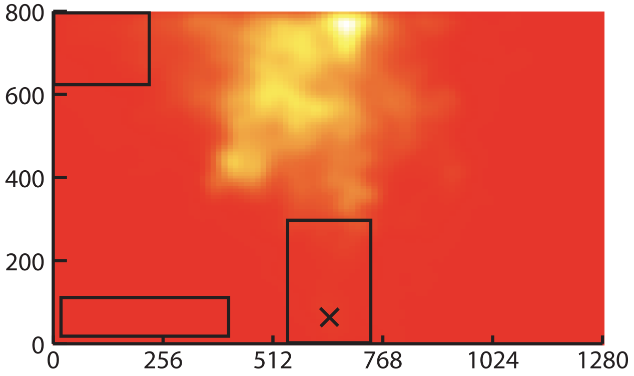


Figure 6: Operator's gaze on the screen distribution, in pixel coordinates. The top-left rectangle represents the 2D video window, and the bottom-left one the text display. As a reference, the cross and the center rectangle represent the base and model display location of the robot in the default initial viewpoint configuration.

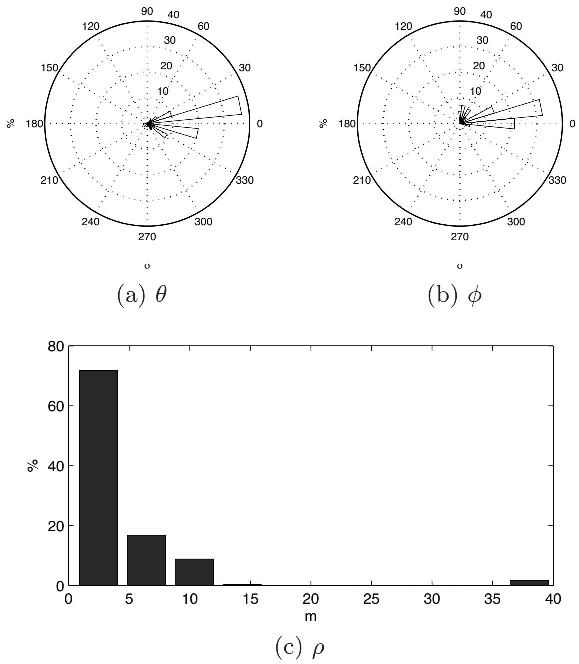
in tight navigation situations like going through doorways, situations that happened less frequently than other tasks. Finally, the ρ distribution indicates that ego- or closely located exocentric positions ($\rho < 5$ m) were preferred. In a few occurrences, operators positioned the virtual camera at more than 30 m to get a full top-down map view of the traveled path.

Changes in either θ, ϕ or ρ occurs 78.2% of the time. Our initial hypothesis was that operators familiar with videogames might adjust the viewpoint more often, since it is a feature available in most 3D games. However, this was not the case: the differences between the two groups are too small to be significant. Operators playing videogames at least one hour per week spent 76.6% of their time using the same viewpoint. This behavior was observed 80.1% of the time with the other operators.

4.3 Control and Task Accuracy

The average time spent while the robot moves (translation or rotational motions) is 34.2%. The average and median translational velocities when moving are 64.9% and 71.0% of the maximum velocity. Furthermore, five operators had their median linear velocities at 100%, meaning they spent at least half of the time in movement at full velocity. This coincides mostly with operators not often exposed to videogames and having difficulties using the analog joysticks for accurate movements. While the robot was in translation, rotational commands were given 54.9% of the time. This implies that operators were confident enough to change the robot's direction without pausing it first. The cases when rotational commands were given without translational motions represent 22.4% of the operational time.

Out of 13 participants, seven managed to complete the tasks without encountering any obstacles, four hit directly or grazed one or two objects, and two hit more than five obstacles. Three operators missed a cone and one counted the same cone twice. Estimating distances with the interface remains difficult, however. On average, operators overestimated distances by 32%. Overestimations occurred 77% of the time and were not consistent between operators. Only one underestimated both distances during its course. Many display factors such as the field of view have been shown to influence distance estimations in virtual environments [18] and would need to be fully assessed in further studies before formulating any conclusions on the distance estimation performance of our system.



(a) θ

(b) ϕ

(c) ρ

Figure 7: Virtual camera position distributions in terms of percentage of time spent in distinct positions. In (a) and (b), robot's heading is at 180°. The optical axis always cross the center of both graphs.

4.4 LASSO Classification

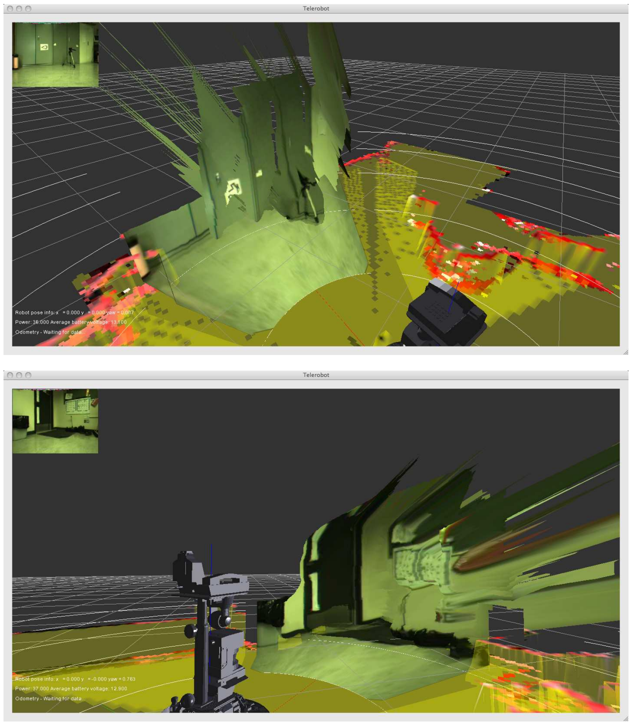
The amount of utterances varies greatly from operators to operators, as in [5]. In our case, it could be as low as 7 or high as 61 per operator, for a total of 311 utterances. Table 4.4 shows that most of the negative utterances were tallied in the Surroundings category and were related to the state of the stereoscopic projection. Operators would complain that they could not figure out what was going on in front of them, until they switched off this projection method and went back to laser-based projection. Doing so sometimes generated a positive utterance in the same category. Figure 8 shows a scene with minimal distortion (top), visible as lines extending to infinity, in reconstructing the cabinets' doors. The bottom screenshot is more distorted: while most of the door stays flat, superfluous displacement is visible in other portions of the scene. Overall mission awareness was usually high, and most positive remarks were related to updating the status of a task, e.g., "I now have seen 3 cones".

4.5 Usefulness

Table 2 suggests that the most useful feature in our interface is laser mapping and extrusion. The operators considered this modality as the only way to make sure that the

Table 1: LASSO classification

Awareness Type	% Positive	% Negative	% Neutral
Location	22.19	2.89	74.92
Activities	6.75	4.82	88.42
Surroundings	59.16	14.47	26.37
Status	15.76	8.36	75.88
Overall Mission	41.48	5.47	53.05



```
Robot pose info: x   = 0.000 y   = 0.000 yaw = 0.007
Power: 36.000 Average battery voltage: 13.100
Odometry - Waiting for data...
```

```
Robot pose info: x   = 0.000 y   = -0.000 yaw = 0.763
Power: 37.000 Average battery voltage: 12.900
Odometry - Waiting for data...
```

Figure 8: Stereoscopic projection screenshots.

robot was not colliding with obstacles. The display of the robot's model in conjunction with the laser range data was identified by many as essential. Another essential feature was the 2D video window, where many operators mentioned that they used it to confirm the nature of objects observed using either of the projection methods. This concurs with the data from Figure 6. Some operators however complained of the small size of this window, and would have like to have a quick way to temporarily display the unprocessed video stream in the main window.

The adjustable viewpoint feature was also appreciated. Being able to quickly tilt the viewpoint to obtain a close, top-down view of the robot's surroundings as perceived by the laser range finder was considered to be essential for obstacle avoidance, especially while crossing doorways. However, one operator would have liked to be able to reconfigure the gamepad's layout by reassigning buttons or inverting the direction of the effect of the analog pad on the viewpoint's angle. This would be straightforward to implement in a manner similar to videogames and should be considered in future work.

Table 2: Usefulness average, from 1 to 5, according to operators after their trial run.

Laser Mapping and Extrusion	4.7
2D Video Window	4.2
Adjustable Viewpoint	4.0
Laser-based Projection	4.0
3D Stereoscopic Projection	1.9

Laser-based projection was also perceived as useful and, understandably, more than the stereoscopic projection. However, the stereoscopic projection was still turned on 49.1% of the time. Operators' ratings were often accompanied by a comment on how stereoscopic projection could certainly be helpful in figuring out the general structure of a room and would be really useful if the projection was more stable and less distorted. Both of our projection techniques provided good results when disparity data were sufficiently complete. In practice however, this proved to rarely be the case, which affected mostly stereoscopic projections. For images presenting little texture information, like a plain-colored wall, the disparity estimation algorithm implemented in Videre's FPGA simply could not provide enough useful information. It however performed fairly well for high-contrast zones like door frames over the same light-colored walls.

Finally, an unexpected remark was made concerning auditory feedback from the teleoperated robot. Even though the operators could not see the course, they could hear the robot moving for the first half of the trial. Some operators commented that they would have liked to have auditory feedback confirming that the robot is moving in areas where the robot could not be heard. Voice feedback communicating navigation decisions of a supervised, semi-autonomous robot has been noted as being effective and appreciated in similar situations [13]. Auditory feedback modalities would therefore be a nice feature that could be easily integrated in our interface.

5. CONCLUSION AND FUTURE WORK

This paper presents a complete proof-of-concept demonstration of a new ego-/exocentric 3D interface that combines laser range finder data with 3D video information. We have shown that it is possible to efficiently combine in real-time laser range finder data with projection of 3D video data. Our interface's adjustable viewpoint feature gives flexibility to operators in selecting what reveals to work better in specific situations.

In future work, our system will be extended by adding an improved stereo disparity method, adding user-specifiable viewpoint presets and auditory feedback of the robot while in motion. The revised interface will then be a versatile tool to conduct rigorous evaluations of display modalities and visualization features with a larger number of operators and in conditions involving moving obstacles. It will also be used to conduct usability studies with novice operators in real telehealth homecare assistance tasks. Furthermore, we are currently extending our Kd-ICP mapping technique to reconstruct the navigated environment in 3D as in [17].

6. ACKNOLWEDGEMENTS

The authors gratefully acknowledge the contribution of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada Research Chairs (CRC) and the Fondation of the Université de Sherbrooke for their financial support.

7. REFERENCES

[1] J. A. Adams. Critical considerations for human-robot interface development. In *AAAI Fall Symposium: Human Robot Interaction Technical Report FS-02-03*, pages 1–8, November 2002.

[2] M. Baker, R. Casey, B. Keyes, and H. A. Yanco. Improved interfaces for human-robot interaction in urban search and rescue. In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, volume 3, pages 2960–2965, October 2004.

[3] J. F. Blinn. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques*, pages 192–198, New York, NY, USA, 1977.

[4] J. Cooper and M. A. Goodrich. Towards combining UAV and sensor operator roles in UAV-enabled visual search. In *Proceedings of the 3rd ACM/IEEE international conference on Human-Robot Interaction*, pages 351–358, New York, NY, USA, 2008.

[5] J. L. Drury, B. Keyes, and H. A. Yanco. LASSOing HRI: analyzing situation awareness in map-centric and video-centric interfaces. In *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction*, pages 279–286, New York, NY, USA, 2007.

[6] J. L. Drury, J. Scholtz, and H. A. Yanco. Awareness in human-robot interaction. In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, volume 1, pages 912–918, October 2003.

[7] D. Hestand and H. A. Yanco. Layered sensor modalities for improved feature detection. In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, volume 3, pages 2966–2970, October 2004.

[8] B. Keyes, R. Casey, H. A. Yanco, B. A. Maxwell, and Y. Georgiev. Camera placement and multi-camera fusion for remote robot operation. In *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, August 2006.

[9] D. Labonté, F. Michaud, R. Cloutier, M.-A. Roux, D. Létourneau, and P. Boissy. Telerehabilitation via a mobile robot. In *Proceedings of Virtual Rehabilitation*, August 2008.

[10] F. Michaud, P. Boissy, H. Corriveau, A. Grant, M. Lauria, D. Labonté, R. Cloutier, M.-A. Roux, M.-P. Royer, and D. Iannuzzi. Telepresence robot for home care assistance. In *Proceedings of the American Association for Artificial Intelligence Spring Symposium on Multidisciplinary Collaboration for Socially Assistive Robotics*, pages 50–55, March 2007.

[11] P. Milgram and H. Colquhoun. A taxonomy of real and virtual world display integration. In *Mixed Reality: Merging Real and Virtual Worlds*, ch. 1. Y. Ohta and H. Tamura (Ed.), Springer, June 1999.

[12] C. Nielsen, M. Goodrich, and R. Ricks. Ecological interfaces for improving mobile robot teleoperation. *IEEE Transactions on Robotics*, 23(5):927–941, October 2007.

[13] X. Perrin, R. Chavarriaga, C. Ray, R. Siegwart, and J. del R. Millán. A comparative psychophysical and eeg study of different feedback modalities for HRI. In *Proceedings of the 3rd ACM/IEEE international conference on Human-Robot interaction*, pages 41–48, New York, NY, USA, 2008.

[14] F. Pomerleau, F. Ferland, and F. Michaud. KD-ICP for Fast and Robust Map Registration. Submitted to the *IEEE International Conference on Robotics and Automation*, 2009.

[15] B. Ricks, C. W. Nielsen, and M. A. Goodrich. Ecological displays for robot interaction: A new perspective. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2855–2860, 2004.

[16] J. Scholtz, J. Young, J. Drury, and H. Yanco. Evaluation of human-robot interaction awareness in search and rescue. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2327–2332, April-May 2004.

[17] S. Thrun, C. Martin, Y. Liu, D. Hahnel, R. Emery-Montemerlo, D. Chakrabarti, and W. Burgard. A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots. *IEEE Transactions on Robotics and Automation*, 20(3):433–443, June 2004.

[18] B. G. Witmer and P. B. Kline. Judging perceived and traversed distance in virtual environments. *Presence: Teleoperators and Virtual Environments*, 7(2):144–167, 1998.

[19] H. A. Yanco, M. Baker, R. Casey, B. Keyes, P. Thoren, J. L. Drury, D. Few, C. Nielsen, and D. Bruemmer. Analysis of human-robot interaction for urban search and rescue. In *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, August 2006.

[20] H. A. Yanco and J. L. Drury. ”Where Am I?” Acquiring situation awareness using a remote robot platform. In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, volume 3, pages 2835–2840, October 2004.

[21] H. A. Yanco and J. L. Drury. Rescuing interfaces: A multi-year study of human-robot interaction at the AAAI robot rescue competition. *Autonomous Robots*, 22(4):333–352, 2007.

[22] H. A. Yanco, J. L. Drury, and J. Scholtz. Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition. *Journal of Human-Computer Interaction*, 19(1-2):117–149, 2004.