A Multi Modal Web Interface for Task Supervision and Specification

Benoît Moreau, Nicola Tomatis, Kai O. Arras, Björn Jensen, Roland Siegwart
Swiss Federal Institute of Technology Lausanne (EPFL)
CH-1015 Lausanne EPFL, Switzerland

ABSTRACT

In this paper we present a multi-modal web interface for autonomous mobile robots. The purpose of this interface is twofold. It serves as a tool for task supervision for the researcher and task specification for the end-user. The applications envisaged are typical service scenarios like remote inspection, transportation tasks or tour guiding.

Instead of post-processing a huge amount of data gathered and stored during operation, it is very desirable for the developer to monitor specific internal variables of the target system in real-time. Sensory information on several levels of abstraction are visualized using state-of-the-art web technology yielding a plug-in-free interface which can be viewed with a standard browser. It provides multi-modal information in several representations: off- and on-board vision, laser and odometry. This tool proved to be indispensable in the developing phase of navigation algorithms for localization, map building, obstacle avoidance and path planning.

Modern guidelines for ergonomic interface design like context-sensitive popup menus or clickable goal specification make the interface very intuitive for the end-user.

Its practicability is extensively demonstrated in the 'Computer2000' exhibition event, where during 4 days the robot was remote controlled in a fully autonomous mode by visitors of the tradeshow using this interface.

Keywords: Multi-modal interface, web interface, task supervision, task specification, mobile robots, remote inspection.

1. INTRODUCTION

Since its initiation at CERN in 1992, the World Wide Web has defined a technological revolution providing a standard graphical interface to the internet [1]and with a continuous exponential grow in the number of users. This technology has therefore turned out to be an answer to the desire of remotely operated mechanisms, which are still a research topic for inhospitable environments such as space, undersea, radiation sites etc. Since 1994 some projects exploiting the World Wide Web for robot interfacing have started[2,3]. Many other have followed[4,5,6] and their interfaces have been improved and fine-tuned for user friendly and effective remote operation on distant machines. The first problem encountered by doing this was the limited bandwidth of the Web. Data from the remote machines has to be optimized to have a maximum perceptual bandwidth while maintaining bounded data transfer [6,7]. Another problem arising with the Web is synchronization. This has to be faced actively because important delay due to network communication causes difficult control and delayed reaction of the machine and therefore security danger for the machine neighborhood. Complex structures have been proposed to solve this issue, but nothing can change the fact that the distant machine will react with some delay to the user command and that the user will see the results of his action with further feedback delay. Most recently, on board intelligence has been used to develop a collaborative system, based on man-machine dialogs[8]. It uses a complex, message based architecture.

In this project both the motivation and the approach to robot interfacing via the Web differ slightly with respect to the above-mentioned works. Firstly the motivation comes directly from mobile robot research: For validation experiments the robot behavior was evaluated by post-processing log files downloaded after the end of the experiment. Data were visualized by means of software like Matlab®. This is an asynchronous and time consuming task, which tends to complicate the theory-programming-validation cycle and therefore slows down research. Because of this the goal of the project was to develop a supervision tool allowing continuous and synchronous system control, without reducing the robot autonomy or modifying its behavior and results in general. Furthermore the approach is different because in this case the intelligence is not given by the user, it is instead directly embedded in the machine, a fully autonomous and self-contained mobile robot. The synchronization problems simplified and no complex watchdog methods on actions are needed, allowing a purely HTML, JavaScript and CGI implementation.

Proceedings of SPIE Vol. 4195: Mobile Robots XV and Telemanipulator and Telepresence Technologies VII, Boston, USA, 5 - 8 November, 2000.

2. DESIGN ISSUES

2.1. The robot systems

In our lab, three robots are in use for research purposes. Pygmalion, Donald and Daffy Duck are fully autonomous mobile robots. They are VME based systems with a six axis robot controller carrying a PowerPC processor at 300 MHz. By using industry standards like Ethernet, VME, IP, PMC and CAN-busses, they provide extensibility for further cup's, periphery and sensors like PCs with touch panel, infrared communication systems, ultrasonic sensors, etc. Besides wheel encoders, their sensory system includes two sick LMS200 laser range finder and a CCD camera. Furthermore Pygmalion has eight tactile zones at two different heights that are used as bumpers. All the robots are connected via radio ethernet only for control and feedback; calculations are done on-board and on-line.

The operating system is XO/2, an object-oriented, hard-real time system software and framework, designed for safety, extensibility and abstraction[9]. It takes care of many common issues faced by programmers of mechatronic products, by hiding general design patterns inside internal mechanisms or by encapsulating them into easy-to-understand abstractions. With this, the researcher is relieved from many computer-science aspects and can better focus his attention to the actual problem to be solved. Furthermore careful handling of safety aspects has been the criterion by which the system has been crafted. This compact embedded operating system (less than 2 MB) is delivered with support for standards like Telnet (server), FTP (server), TFTP (client and server), STMP (client), HTTP (client and server), etc.

When travelling around our robots use, beside all the low-level controllers and drivers, the encoders and the exteroperceptive sensors (laser scanner and CCD camera) for multiple purposes:

1. Encoders are employed for dead reckoning. By counting the wheel rotation a raw estimation of the robot position is maintained (odometry).
2. Laser scanner:
   - It is used for obstacle avoidance. The robot avoids obstacles represented by the raw measurements of the scanner by planning a collision free path reaching a predefined goal.
   - It is employed for localization in the environment. Segments are extracted from the raw data and matched with infinite lines, which are either compared to those representing the map of the environment or used for topological localization.
3. The CCD camera is used for localization in the environment. Vertical lines are extracted and compared to the map or combined with data from the laser scanner for topological navigation.

These tasks run in parallel into the embedded system. They were a research topic for [10], have been widely tested in [11] and are still matter of research for map building and topological navigation.

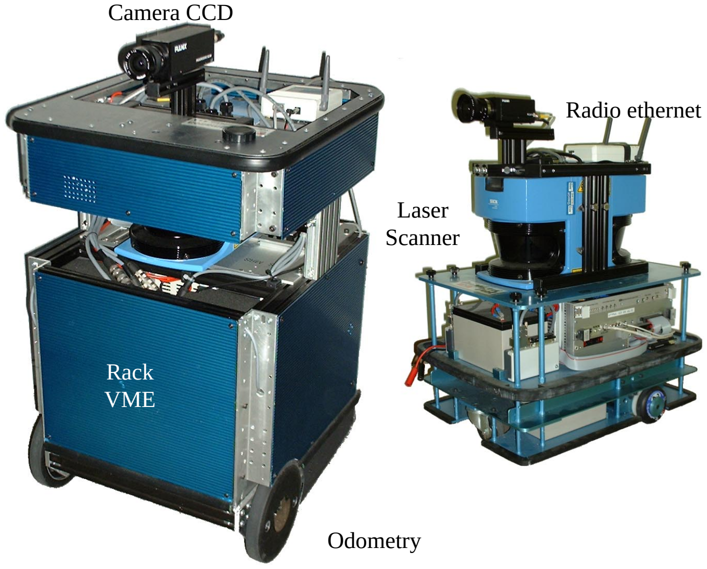

Figure 1. Robot systems (Pygmalion – SmartRob)

2.2. Data feedback for task supervision

Testing algorithms like localization and obstacle avoidance on an autonomous self-contained robot[9] requires a mean for the researcher to check the algorithmic reactions to the machine perception of the world. Now, because of the characteristics of such an embedded system, the perceived data are also embedded in the mobile vehicle and the on-board acquisition-transfer-processing cycle remains hidden to the outside until all these data are transferred to a local PC for analysis. This can be done by tracing the robot position (odometry), saving all the raw data from the sensors, the extracted features and the results of each algorithm and transferring this information when an experiment is finished. Off-board analysis can then be done easily by visualizing and interpreting the experimental data and results. Nevertheless this procedure has several disadvantages:

- The correspondence between the behavior of the robot and the data, which caused this behavior, is difficult to identify.
- Critical states of algorithms that may cause a failure cannot be detected immediately before and are therefore difficult to isolate and correct.
- Crashes of one or more processes can cause a loss of information which would be important for the off-line analysis

On-line supervision is therefore not a luxus, it is instead an important tool for speeding up the advances in application like mobile robotics research by permitting on-line individuation of characteristics and behaviors of the tested approaches. However, machine perception differs widely from human perception (i.e. on-line visualization of bits or numbers in general remains unsatisfactory). In the best case the researcher would have a global, simple and user-friendly view of the complete robot state at each time. This is practical unfeasible since such a complex embedded system has too much information for human perception. Nevertheless graphical visualization permits to increase the perceptual bandwidth[7] of the world-machine-human channel, allowing the researcher to better interpret the machine behavior by comparing machine perception with its own one (Figure 2). Therefore a graphical representation of the raw data from the sensors, the predicted, extracted and matched features and the robot position in the known environment is well suited for this purpose.

2.3. Control for task specification

By performing public presentations and common experiments with distant labs, people reactions demonstrate how they were interested but frustrated at the same time by looking at such a system. Obscure inline commands were used to control the robot and the only feedback was the behavior robot and some text output. This was not satisfying for people who were not familiar with this development and operating system. Therefore including task specification in a graphical feedback interface for making the results of the research in robotic accessible to potential end-users became a major aspect of the development. Furthermore, for distant users, it was difficult to understand what was happening without having a view from outside the robot.

Therefore a mean for controlling the robot, which was not a basic aim of the project, has been developed. Defining a navigation goal from a graphical interface showing the known environment is a simple approach for letting the user interact with the mobile vehicle in a user-friendly way. The graphical representation presented in the last section permits the user to understand what happens to the mobile robot, how it can remain localized with respect to the environment, how it reacts to the dynamic of the environment and so on. Visualization of the raw data allows seeing the neighborhood of the robot. This can be exploited to define local goals near the current robot position. Furthermore external cameras permit distant users to see the robot behavior and its environment.

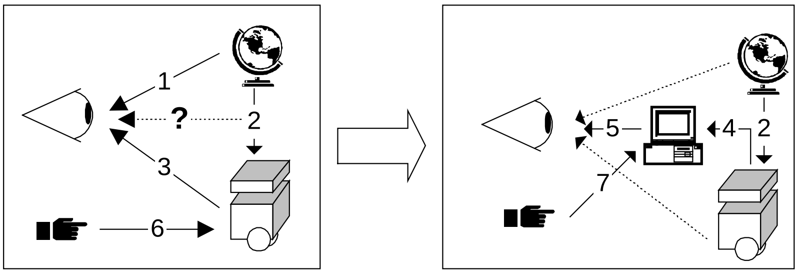

Figure 2. To understand what is the robot vicinity perception, knowing perfectly on board algorithms, having access to sensor data and moreover being able to do simultaneous correlation between 3 and 4-5 is necessary. Face to that impossibility, only high level conjunctures were done. For that, an interface proposing already correlated data has been introduced. 1: Human perception of the real world. 2: Machine perception of the real world. 3: Human perception of the robot behavior. 4: On-line transfer of the machine perception and machine states. 5: On-line visualization of the machine perception and machine states. 6: Human commands via in-line text commands. 7: Visual task specification.

3. IMPLEMENTATION

To reply to all those needs, we have developed a platform independent web interface, which can be viewed with the standard Netscape® browser. This interface shows where the robot is, what it is doing and even more what is its environment perception. On the user action side, the interface allows to control the robot from high abstraction level commands down to sub-centimeter precise motion.

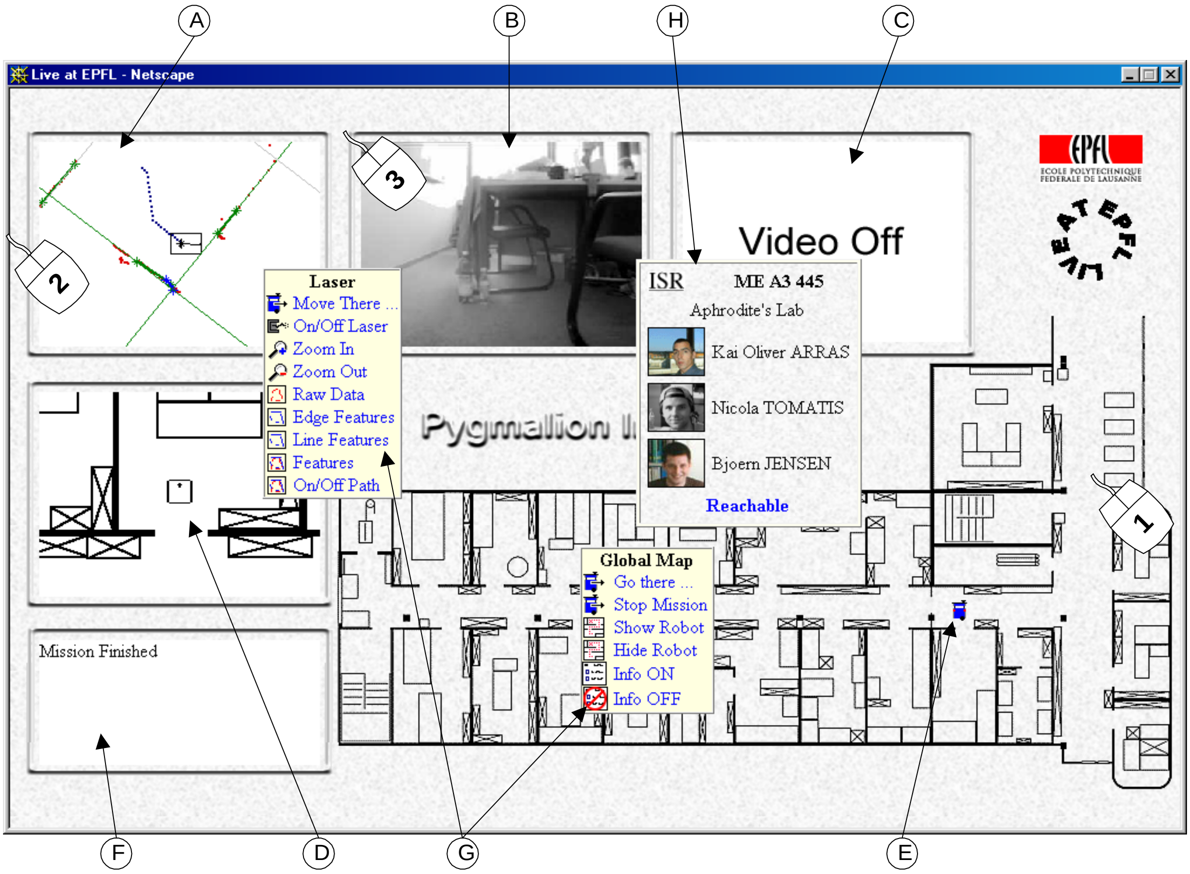

Figure 3: The web interface. A: Multi sensor localization monitor. B: On board video. C: External video. D: Local robot position (x,y,θ). E: Global robot position (x,y). F: Message window. G: Popup menu on each window to access corresponding functionality. H: Office information popup menu on global map. Numbered mice are possible way to control the robot. Mouse1: Set a room as new global goal. Mouse2: Set (x,y) as new local goal. Mouse3: assign new orientation (θ).

As shown in the Figure 4: Global Structure, the system is based on a compact architecture. To avoid communication overhead due to disconnection/reconnection proper to http data transfer, the use of Netscape "push" technique[12] has been generalized for continuous data flow as video. Large dynamic control of the interface from the embedded web server is ensured with a very low bandwidth by means of the use of a hidden window in which the robot can directly push JavaScript[13]. This code is interpreted as soon as a page is totally pushed (page: <HTML></HTML>). Due to the fact that the data flow is continuous, the autonomous machine can control the interface at any time, on its own initiative. On the other side, data is simply displayed into a dynamic interface, which is, as stated before, controlled and configured by the distant system. Interface functions are coded to transform data stream into simple procedure calls. That reduces the flow and simplifies the development on the server side. Only header functions are needed to use the interface. For the other streams, all bandwidth consuming data are commonly preprocessed before transfer as explained in concerning part.

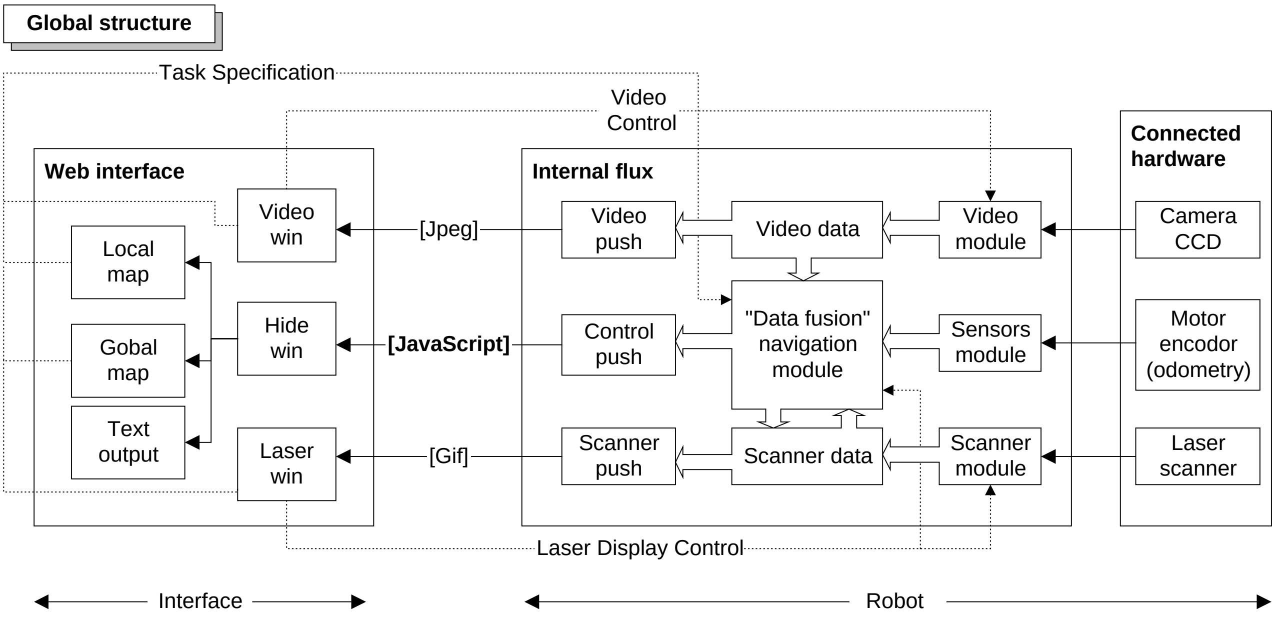

Figure 4: Global Structure

3.1. Task supervision

This section describes the interface part, which has been explicitly developed as a data feedback tool for research purpose.

3.1.1. Multi sensor localization monitor(Figure 3, A)

This feedback window is based on a 2D graphical representation of the laser scanner measures completed with processed data. The GIF[14] format image created from the data is scalable to represent the neighborhood of the robot from 1 meter to 10 meters width (Figure 7). A proportionally scaled robot is added. Starting with these main features, two modes are available. One shows the environment with a metric grid (Figure 5), and the other visualizes explicitly the results needed in that research context, like the predicted, observed and matched features from both the laser scanner and the CCD camera (Figure 6). On both of those representations, information from the navigation module is available as a graphical representation of a local path planed.

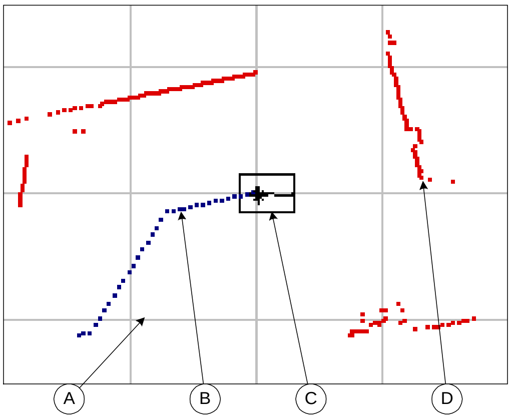

Figure 5 A: Metric Grid, cell unit is the meter. B: Path planed. C: The proportional robot representation. D: Laser scanner raw data.

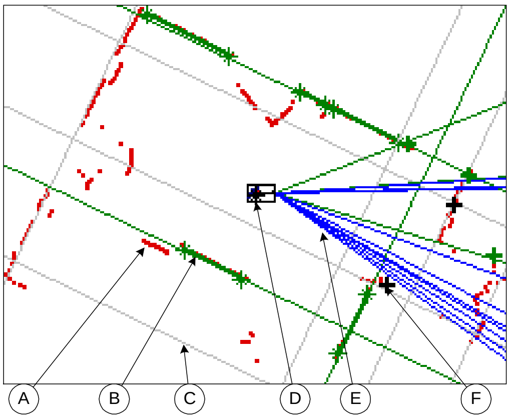

Figure 6. A: Laser scanner raw data. B: Extracted lines (green for recognized, blue for other). C: Model lines from the a priori map. D: The proportional robot representation. E: extracted vertical edges (green for recognized, blue for other). F: Model edges from the a priori map.

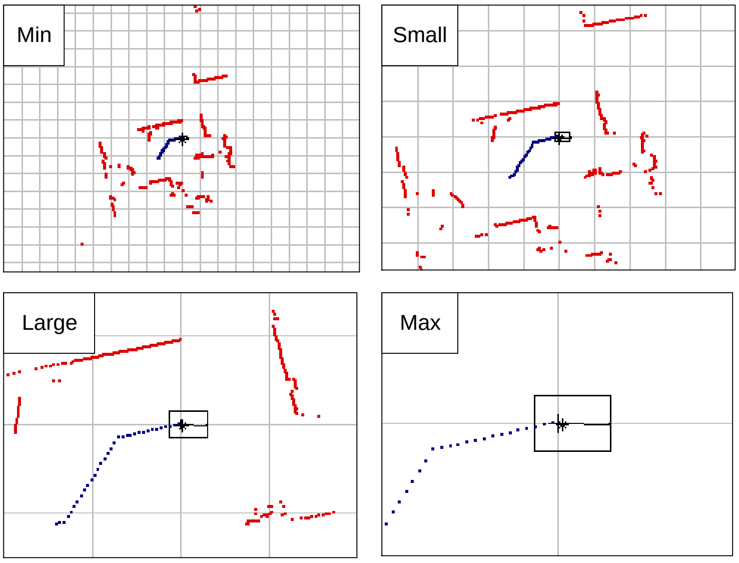

Figure 7. System zoom abilities

3.1.2. On board video(Figure 3, B)

There are three kinds of processes applied to the video to obtain a continuous image flow: Two pre-processes and a post-process. Firstly the picture is white balanced to obtain an optimized contrast and then encoded in Jpeg[15] format. Data transfer is again based on Netscape push technique. On the interface side, images are automatically scaled by the browser to fit the allocated layer space.

The Jpeg[16] encoder has been developed in the context of this project and has been further optimized[17]. Optimizations are twofold. The first is the used of two consecutive rapid "1D-DCT"[18] instead a single "2D-DCT". The other one simply stops the second "1D-DCT" when most important information are compressed. This feature is not of primary interest for off-board image processing, but in this case, it becomes relevant since the generation of these images is on a on-board self-contained system that has to share its resources between many time critical tasks. To use the encoder, two parameters are required. They both modify the image quality but do not intervene in the same part of the process. The first parameter is the one defined by the Jpeg standard. It reduces the file size proportionally to its resolution. The second parameter defines when the second "1D-DCT" should stop.

3.1.3. Local and global robot position (Figure 3, D, E)

Both global and local maps use the same layer repositioning technique. The global one slides a small robot picture over the image of the whole environment map, while the other move a bigger image of the map, through a fixed visibility area, under a machine image layer. The local one does also picture flip on the top layer to display orientation of the robot by rotating the picture representing it.

The bandwidth consumption is then very low, only (x,y,θ) are pushed. (x,y) is the metric instant thought position of the robot in the lab, θ is its orientation. The size of the robot environment scale, resulting in the corresponding map, is defined in the web code; conversions are done online.

3.1.4. Message window (Figure 3, F)

An internal JavaScript function has been coded to be used by the robot, via the pushed data stream, and by the interface. It redirects an input string on a pointed layer. This technique allows text only data display from multiple sources. It is actually used when all local components are not available and to display mission states.

3.1.5. Performances

To ensure effective supervision of the robot behavior, the feedback delay has to be minimized, while the refresh rate has to be maximized. This section analyzes the characteristics of the proposed implementation and shows the applicability of such an approach for the mobile robot systems presented here.

		Laser GIF picture		Video JPG picture	
		Raw data	Localization data	Low resolution	High resolution
Time	Acquisition time (ms)	20	40	100	100
	Compression time (ms)	100	100	80	300
	Picture / sec	8.5	7	5.5	2.5
Size	Size (Bytes)	1 457	2 973	2 165	5 294
	Round Size (Kb)	12	27	18	43
Flow	Possible flow (Kb/s)	102	**189**	99	**107.5**

Table 1. Data flow statistics

The actual wireless network allows 3Mb/s. Table 1 shows that the maximum required bandwidth is about 300 Kb/s (laser + video). This proves that the approach presented here is compatible with the network hardware provided by the systems in use. Nevertheless, when the robot information is accessed by a remote Internet browser, unpredictable flow congestion can still take place.

3.2. Task specification

In this section features allowing robot control via the web are described. For this, left clicking on the interface's windows can activate pop-up menus. These menus permit to define local and/or global navigation goals. As soon as a goal is defined, further user assignments are ignored until the current mission is accomplished.

3.2.1. Local goal specification (Figure 3. Mouse 2, 3)

Local goals can be assigned in the localization monitor and in the on board video windows. To easily define a navigation goal under account of the physical dimensions of the robot and objects in its vicinity, a representation, which displays true metric information, is needed. The raw data mode is therefore used, but the technique presented here is fully compatible with all other modes. By clicking on the display, the (x,y) cursor position is transformed into a global goal by a simple metric conversion.

$$\mathrm{GlobalGoal} = \left[\left(\frac{\mathrm{VisibleAreaWidth}}{\mathrm{DisplayWidth}}\right) * \left(\mathrm{LocalClickPosition}\right)\right] + \mathrm{RobotGlobalPosition}$$

Due to maximal definition available, the goal assignation has a maximal precision of

$$\mathrm{MaxPrecision} = \left(\frac{\mathrm{MinVisibleAreaWidth}}{\mathrm{DisplayWidth}}\right) = \left(\frac{200}{240}\right) < 1\mathrm{cm}$$

which is approximately our navigation precision[11].

A local goal definition is also possible through the on board video window. It allows setting a new θ for the robot. So turning to get a round look is possible. Having a vicinity global vision simplifies the end user comprehension and interpretation of the laser representation. The new angle is set by the distance from the middle of the picture to the click position on the x axis. It is proportional, and the max θ depends on the opening of the camera optics (a).

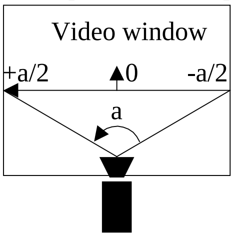

Figure 8. Link on camera view angle and video window click ability

3.2.2. Global goal specification (Figure 3. Mouse 1)

On the global map, rooms are implemented as active zones (HTML link map). For each, on “Mouse Over” event, a pop up menu appears. It displays information about the chosen place, such as the number, photo and name of people working in it, and if the robot can reach the place (REF figure 3: (H)). On “On Click” event the functionality pop up menu appears, which allows to assign corresponding office as a new global goal. A list of the coordinates associated with places is stocked on board, and only room numbers are transferred to the robot through a simple CGI.

3.2.3. External video (Figure 3, C)

The same push technique is used for multiple cameras from external HTTP server to the interface. Cameras are 640x480 web-cams, but any kind of hardware can be use. Those servers are installed on several desktops computer in different rooms. Employing existent machine and low cost cameras allows a large deployment of the system. The video is pushed from one active source into the concerned window (Figure 3, C). Switching the source is simply done by left clicking on the display and then choosing another camera from those appearing in the list. To know which one is active, a camcorder, representing the video source, is drawn on the global map.

4. EXPERIMENTAL RESULTS

As stated in the introduction, the use of this interface as a tool for task supervision has received particular attention. We first present the possibilities the system provides for monitoring the internal states of two different localization systems, one being Kalman filter-based and the other POMDP-based. Finally, we present the ‘Computer2000’ event, an extensive test of the overall navigation system, where during four days visitors of the ‘Computer’-tradeshow remote-controlled Pygmalion by means of this interface.

4.1. Application as a localization monitor

It is evident that knowing the internal states of a localization system and supervising them in real-time is extremely useful when trying to understand its functioning and to obtain a feeling for its performance. This is particularly the case if localization is done with an unimodal approach where a single probability distribution is maintained and tracked as it is the case with the Kalman filter[11]. The robot can go lost if the truth departs from the modeling assumptions fast with respect to the cycle time, that is, when the tracker looses its track. This can happens for example after collisions or after a continuous lack of sensorial information in case of people fully surrounding the robot for a long period of time.

With this interface, the researcher can illustrate raw data, modeled features in the map (in our case infinite lines and vertical edges), the current observations (line segments and vertical edges) and their matching information (see fig. 3). Raw data is drawn in red, unmatched modeled infinite lines are drawn in light gray, unmatched modeled vertical edges are drawn as black crosses, unmatched observations are drawn in blue and matched pairings are indicated by replacing the default colors with green for both, predictions and observations. With this functionality the extraction algorithms for lines and vertical edges can be observed on-the-fly as well as the behavior of the Kalman filter.

Similarly, the interface has been used to supervise the robot behavior in the development phase of a new localization approach[19]. In this case the robot navigates in the large (in our lab that means navigation in hallways) by means of a topological based approach. It does not know its exact metric position. Instead of this, it has a probability distribution over all the possible places in the environment. This is often referred to as a multimodal approach. When the mobile system reaches a place where it has to accomplish a task, the navigation switches to the above-presented unimodal Kalman filter approach in order to have a precise estimate of its position. For the development of this hybrid approach the advantages of graphical web interfacing were a full task supervision permitting to see where the robot had an unexpected behavior and to check when the experiment had to be stopped for security reasons. More precisely, the internal state of the position tracker was visible by comparing the raw data perception of the sensors with the probability distribution over the places. Furthermore, when changing from topological to metric, which is a very critical transition for such a method, it was very simple to detect if the robot successfully initialized the Kalman by looking at the matched features in the multi sensor localization monitor.

Furthermore, local path planning information is illustrated. Based on the raw range information from the laser scanners, a collision-free path is generated in a local grid around the robot. This path can be depicted in real-time, during motion or at standstill, varying the obstacle configuration. Besides the use of this functionality for research and education purposes, the interface turned out to be an extremely helpful tool in daily development practice as it allows almost immediately to detect bugs in algorithms and code. During the development phase large-scale experiments with the real vehicle can be safety-critical. Here it serves as a safety tool for operator intervention. Critical states that could hinder the security of the neighborhood of the mobile vehicle was detectable by looking at the robot position and its planned path.

4.2. The ‘Computer 2000’ event

The ‘Computer’ tradeshow is an annual fair for computer hard- and software at the Palais de Beaulieu exposition center in Lausanne, Switzerland. Our laboratory was present during the four days, May 2nd to May 5th 2000, giving visitors the opportunity to control Pygmalion by means of the web-interface. The robot itself was at EPFL, in the environment illustrated in fig. 3. It is 50 x 30 meter in size and contains twelve offices, two corridors, the seminar and the mail room.

The ‘Computer 2000’ event was the final testbed for the whole navigation system consisting in the interface, obstacle avoidance, global and local path planning and the Kalman-based localization. We were particularly interested in long-term reliability under application-like conditions of the latter component. Obstacle avoidance and the local path planner are based on laser range data, the global planner operates on a topological graph the a priori map contains. The setup was active during normal office hours with an average of about 7 hours system up time per day. The environment exhibited typical dynamics from people, doors, chairs and other robots, as well as daylight illumination. Several doors open into the corridor (see fig. , right image). Travel speed has been limited to 0.4 m/s since the robot shares its environment with persons, some of them not implied into robotics.

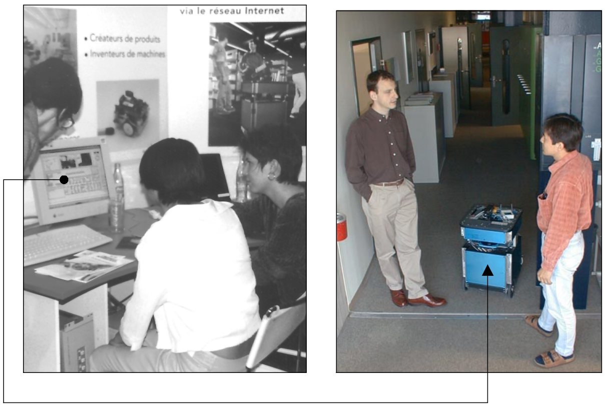

Figure 9. During the four days of the Computer 2000 event, visitors could teleoperate Pygmalion in the corridors and offices of our institute building at EPFL.

The event statistics of Computer 2000 is shown in table . A mission is either a global or a local navigation command from the interface.

Hours of operation	28
Environment size	50 x 30 m
Environment type	office, unmodified
Overall travel distance	5,013 m
Average speed	0.2 m/sec
Travel speed	0.4 m/sec
Number of missions	724
Number of localization cycles	145,433
Number of lost-situation	0
Number of unknown collisions	~ 10

Table 2. Overall statistics for the Computer 2000 event.

A *lost situation* is defined as a mission whose goal could not be attained due to a localization error and which required manual intervention to re-localize the robot. We do not count missions where the robot went lost due to a collision with an invisible object (e.g. glass door or object lower than the beam height of the scanner) and where the robot was already lost (after such a collision). When the vehicle is surrounded by people, when it is in less structured terrain or when odometry delivers inconsistent position estimates (e.g. from driving over uneven floors), it happens that there are no matched features during a certain period. We counted 14 of 724 missions where the robot had no matches during 10 seconds, 21 missions where it had no matches during 5 seconds. None of them required manual intervention during or after the mission. These extraordinary results where further underlined by the positive feedback we obtained from the big amount of visitors. They particularly enjoyed the user-friendliness of the interface, which permits to practically anybody to explore our laboratories at EPFL with a mobile robot.

5. CONCLUSION AND OUTLOOK

The main result of this project is the implementation of a "real web-based" interface. Unlike others, our interface is neither a connected independent software running off-board, nor an executed and then linked program. For us, "real web-based" means that the robot is called through a standard browser terminal, in which it simply displays data. With such a simple interface we demonstrate the advantages of using fully autonomous robots for telepresence: The synchronization problem is simplified by on-board intelligence. The navigation abilities of our autonomous systems in an indoor dynamic environment have been proven by experiments, where navigation goals were defined by non-trained users interacting with the robot and its environment only by means of the web interface. This demonstrates an advantageous perceptual bandwidth transmitted via the limited data transfer of the network.

Future work will focus on the implementation of some new features that should not change the interface architecture: Now the robot is already able to identify when it cannot fulfill a mission. Correlating this information with the defined navigation goal could allow updating automatically the map for redefining a room as unreachable. This would permit to avoid trying to navigate to areas, which are not reachable anymore due to dynamic changes in the environment. Another interesting feature could be to automate the selection of the active external video: The robot knows its position and the relative distance to the external cameras. It could select the external camera in order to maximize its appearance in the environment where it is moving.

ACKNOWLEDGMENTS

We would like to thank Roberto Brega and Felix Wullschleger at the Institute of Robotics, ETHZ, for their valuable support.

REFERENCES

[1] T. Berners-Lee, R. Cailliau, J-F. Groff, B. Pollerman, "World-wide-web: The information universe", Electronic Networking: Research, Application and Policy, 1:2, Westport, Spring, 1992.

[2] K. Goldberg et al., "Desktop Teleoperation via the World Wide Web", IEEE International Conference on Robotics and Automation, Nagoya, Japan, 1995.

[3] K. Taylor, J. Trevelyan, "Australia's Telerobot on the Web", 26th International Symposium on Industrial Robots, Singapore, 1995.

[4] Siegwart R.: Editor of the Workshop Proceedings 'Robots on the Web', IEEE/RSJ International Conference on Intelligent Robots and Systems, Victoria, Canada, 1998.

[5] Robots on the Web, Special Issue of the IEEE Robotics and Automation Magazine, 7:1, 2000.

[6] G. Terrien , T. Fong, C. Thorpe, C. Baur, “Remote Driving with a Multisensor User Interface”, SAE 30th ICES, Toulouse(France), 2000.

[7] B. Reeves, C. Nass, "Perceptual Bandwidth", Communications of the ACM, 43:3, 2000.

[8] T. Fong, C. Thorpe, C. Baur, “Advanced Interface for Vehicle Teleoperation: Collaboration Control, Sensor Fusion Display, and Web-Base Tools”, Vehicle Teleoperation Interface workshop, IEEE International Conference on Robotics and Automation, 2000.

[9] R. Brega, N. Tomatis, K. Arras, "The Need for Autonomy and Real-Time in Mobile Robotics: A Case Study of XO/2 and Pygmalion", IEEE/RSJ International Conference on Intelligent Robots and Systems, Takamatsu, Japan, 2000.

[10] K.O. Arras, N. Tomatis, "Improving Robustness and Precision in Mobile Robot Localization by Using Laser Range Finding and Monocular Vision", Third European Workshop on Advanced Mobile Robots (Eurobot 99), Zurich, Switzerland, 1999.

[11] K.O. Arras, N. Tomatis, B. Jensen, R. Siegwart, "Multisensor On-the-Fly Localization: Precision and Reliability for Applications". To appear in the Journal of Robotics and Autonomous Systems.

[12] Netscape, an exploration of dynamics documents. Http://home.netscape.com/assist/net_sites/pushpull.html

[13] Netscape Javascript guide. http://developer.netscape.com/tech/javascript/index.html

[14] "GIF, Graphics Interchange Format. A standard defining a mechanism for the storage and transmission of raster-based graphics information", CompuServe Incorporated, 1997.

[15] CCITT T.81 Information Technology - digital compression and coding of continuous tone still images - requirements and guidelines

[16] Eric Hamilton, "JPEG File Interchange Format", C-Cube Microsystems, September 1992.

[17] Moreau B., "Controle et Retour d'Informations pour Robot Mobile", Robotic DEA diploma work, Swiss Federal Institute of Technology Lausanne, July 1999.

[18] Loeffler C., Ligtenberg A, and Moschytz G.S., "Practical Fast 1D DCT algorithms with 11 multiplications", ICASSP89, 1989.

[19] N. Tomatis, I. Nourbakhsh, K. Arras, R. Siegwart, "A Hybrid Approach for Robust and Precise Mobile Robot Navigation with Compact Environment Modeling", Submitted to IEEE International Conference on Robotics and Automation, Seoul, Korea, 2001.