3D Virtual Model for Intelligent Space

Péter Zanaty[1], Gábor Sziebig[2], Péter Korondi[1] and Zsolt Frei[3]

[1]Department of Automation and Applied Informatics, Budapest University of Technology and Economics, Hungary, e-mail: korondi@elektro.get.bme.hu

[2]Department of Industrial Engineering, Narvik University College, Norway, e-mail: gabszi@hin.no

[3]Institute of Physics, Eötvös University, Hungary, e-mail: frei@alcyone.elte.hu

Abstract: By virtually re-creating and visualizing an intelligent space (iSpace) geographically far from the original version, one opens up the possibility of remote control and interaction between the real iSpace and the virtual copy. Here we present the required components for such a mixed environment: the actual iSpace in a form of a laboratory built for such experiments, the stereoscopic (real 3D) visualization tool at the far end to reproduce the original iSpace, and the middleware embodied by the computer network that connects the two. To remotely operate in the iSpace one has to overcome the important problem of synchronization that is made difficult by the time delay in the computer network. Here we present our first approach to solve this problem by demonstrating a successful remote control of a mobile robot in this mixed environment.

Keywords: 3D stereoscopic visualization, Intelligent Space (iSpace), motion capture

1 Introduction

In recent years we saw an active development of the emerging research field of spaces that have distributed sensors for monitoring the space. These spaces have various names, such as smart spaces or ambient intelligence, ubiquitous computing, whereas in our research we have been using the name Intelligent Space (hereon iSpace). iSpace is a room (corridor, street, or any other well defined environment), which has distributed sensory intelligence and is equipped with actuators. For a review, see [1].

There are a number of research approaches similar to iSpace. Easy Living Technologies [2] utilizes human tracking technology with the purpose of guessing the intent of users in the space in order to automate and facilitate everyday tasks. The Oxygen Project [3] is a similar research aiming at the development of intelligent environments based on human-centered computation. The Interactive

Workspace Project [4] is yet another attempt in which the possibilities of using ubiquitous embedded sensors and information displays are explored.

Not all use robots as physical agents. A small number of research approaches (for example [5], [6], and [7]) use them the way we do in our iSpace.

Finally, there is a considerable effort of research on sensor networks, ubiquitous computing [8], and mobile robot control, which is also more or less related to the iSpace initiative.

The main objective of an iSpace is to provide services to humans inside the iSpace. Often mobile robots are introduced into an iSpace as actuators. Tracking humans and tracking and controlling robots in iSpace is a complex task, for a review of current approaches see [9].

In this paper we introduce yet another possible extension on the concept that is being developed: remote control and interaction with an iSpace. Sensory data may need to be transmitted to a remote mainframe computer for processing and control data may need to be sent back to agents in an iSpace. Moreover, remote control by humans of a distant iSpace may be necessary and optimal. Central control rooms could help out those in temporary need in iSpaces, without the requirement of the constant local presence of controllers. Such a remote control introduces the problem of time-delay. We connected an iSpace (located in Japan) to a remote location (in Hungary) where a stereoscopic projection system aided the remote interaction with the iSpace in Japan and provided real 3D feedback of a position of a mobile robot in the remote iSpace. We demonstrated by controlling the mobile robot remotely that the problem of time-delay could be solved with a simple approach. The interaction with iSpace was achieved by utilizing motion capturing technologies, which is the most natural way of communicating with a virtual environment.

In § 2 we describe the iSpace (in Japan), in § 3 we outline the stereoscopic projection system (in Hungary), in § 4 we present the required tools for connecting the two above (for remote control, motion capture and visualization), and in § 5 we detail our simple predictor that was successful in overcoming time-delay problems.

2 Actual iSpace – in Japan

To verify the concept of iSpace, an experimental room was established in Japan [10]. The experimental room is shown in Fig. 1. The iSpace consists of eight CCD cameras, an ultrasound positioning system, and two mobile robots (P2-DX) manufactured by ActiveMedia Robotics.

Figure 1
Experimental iSpace at the University of Tokyo, Japan. Many sensors gather data and two mobile robots (red colored on the left hand side of this picture) act as actuators in iSpace.

In our experiments we did not use the CCD cameras, but collected and transmitted data from the ultrasound positioning system. This ultrasound positioning system consists of small sized transmitters located on the mobile robots and about 90 receivers mounted on the ceiling and around the room.

To achieve high positioning and orientation accuracy, and to ensure fast system response, the ultrasound positioning system is coupled to the mobile robot's wheel encoders. We used Extended Kalman Filters (hereafter EKF) [9], both to filter noisy ultrasound data, and to filter wheel encoder data that might be affected by wheel slippage.

To carry out our experiments, this laboratory was connected though the internet to the remote visualization system described below.

3 Visualization Center – in Hungary

The Visualization Center (hereafter VC) at the Faculty of Science, Eötvös University, Budapest, Hungary was founded in 2006. One of the major projects undertaken was a construction of a large stereoscopic projection system with the primary aim of providing a platform for scientific visualization. This stereoscopic system is well suited for the application that is the subject of the current paper. It provides and immersive and collaborative environment for visualization, therefore it is very useful for visualizing an iSpace at a distance.

There are many Cave Automatic Virtual Environments (hereafter CAVEs) in operation today. A CAVE is an immersive virtual reality environment where projectors are directed to three, four, five or six of the walls of a room sized cube, usually from the outside. They provide total immersion for the user as seen in Fig. 2. The CAVE definition, first introduced in 1992 [11] specifies

- high-resolution projectors,
- back-projection systems, and a
- dark environment to minimize possible reflections.
- the user is usually tracked,

and computer generated images based on the position of the user in the cave provide the feeling that objects float in space and can be walked around.

Figure 2
A typical CAVE. There are three back-projected walls (front, left and right) and one that is projected from above (the floor is "down-projected") in this system. Wired tracking and input devices complete the setup.

In the VC at Eötvös University our stereoscopic system consist of only one wall. This is a cost effective solution, requiring less projectors, but still providing a projected image that fills the viewing angle of the observer almost entirely. This system may not qualify as a CAVE, but besides the number of walls, the other characteristics are there: it is built in an underground, flat-black painted dark room and the large screen is back-projected by high resolution projectors. There are actually two projectors, that both illuminate the same wall. They provide a very wide aspect ratio screen (2.36:1) through edge blending. Each projector has a native resolution of 1400 by 1050 pixels. The two images, projected side-by-side, overlap in the center by 320 pixels. Both projectors send information to these overlapping pixels, but projector power is adjusted in this blended region so that total pixel illumination is always constant. The smooth transition between the two projectors renders the overlap virtually invisible, and provides an 2480 by 1050 pixel single image on a 5 m by 2.1 m screen as seen in Fig. 3.

Since this resolution is high, the projection is from the back, and the screen-size is life-like, the user can move close to the image, where the wide screen fills even his peripheral vision. He is almost totally immersed in the virtual environment, thus the system being cost effective in function comes close to what a CAVE does.

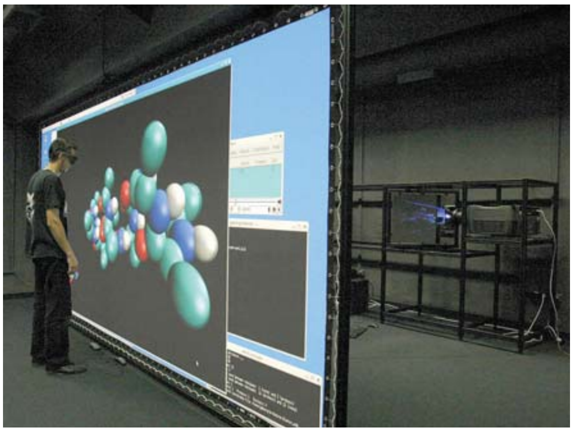

Figure 3
The stereoscopic projection system at the VC of Eötvös University. Active stereo is used to create real 3D images. The screen is large, back-projected and uses a very wide aspect ratio so that the user feels immersed in a virtual environment.

The system provides images from a fixed viewpoint at the moment, but allows the operator to freely move around in the virtually created room. The operator's movement is tracked by a motion capture system, manufactured by Measurand Inc [12]. The motion capture suit is easy to calibrate, wireless and can be operated wherever a wireless network is available. As the system does not require any external devices for operation, highly accurate positional data of the operator cannot be granted. The system has an overall accuracy of 10 centimetres, which is not enough for viewpoint calculations.

To overcome the motion capture's error a simple and robust camera based tracking system is under development.

4 Additional System Components

Humans receive most of their information visually; significant part of the neurons in the human brain is dedicated to visual processing. 3D visualization of information is probably the best way of feedback to the operator. This can be achieved using active stereoscopic rendering with the system described above, but generating one image per eye in order to enhance the depth perception in the virtual world is needed by the computer system.

This type of rendering is supported by special graphics hardware and software, which can draw into four buffers instead of the usual two, and can display eye specific images in sync with the LCD shutter glasses.

In order to utilize the capabilities of the special hardware, we modified and adapted a well-known open source graphical engine for image rendering, called OGRE 3D [13].

The capabilities of the engine we selected are compared to two other leading open source engines in Table 1.

Table 1

Comparison of Image Renderers

Criteria	OGRE 3D	CrystalSpace	OpenGL
Approach	Fully object-oriented	Object-oriented	Native
Documentation	Extensive	Poor	Extensive
Feature set	Good	Moderate	Basic
Performance	Fast	Moderate	Very fast

Object-oriented programming, readability, changeability and maintainability of the source code – which is important in creating a solid framework – were taken into account, when choosing OGRE as our rendering platform. OGRE has also already proven to be industrial-, education-, and commercial-ready in hundreds of successful projects.

There are several projects that use OGRE for 3D visualization, but there is no existing native support for stereo rendering in the framework. In order to enable the stereo rendering mode, the OpenGL rendering subsystem of OGRE has been altered and Quad-Buffer support was introduced [14].

The application developed for providing feedback for the operator uses the OpenGL rendering subsystem and runs in a Linux environment. The virtual equivalent of the iSpace in Tokyo has been created. A 2D rendering of the virtual laboratory can be seen in Fig. 4.

We used the virtual equivalent of the real iSpace in Hungary to project a 3D virtual version of the iSpace on the large screen at the VC. We conducted remote control experiments between Hungary and Japan during March, 2008. The VC stereoscopic screen with the virtual iSpace projected is shown in Fig. 5.

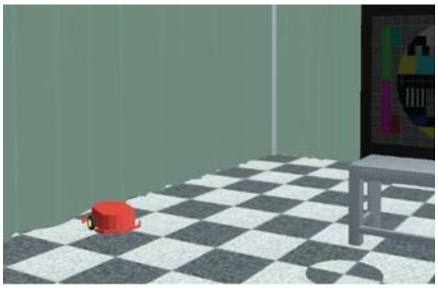

Figure 4
Here a 2D rendering of the 3D model of our iSpace (in Japan) is shown. One mobile robot is pictured here. We only used a single robot in our experiments (see the description in the text).

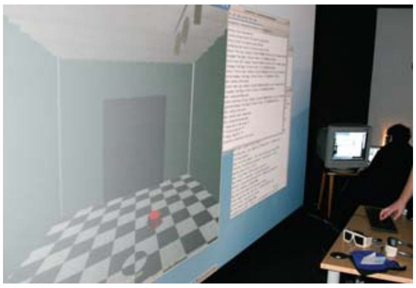

Figure 5
The large stereoscopic projection system with the virtual version of the iSpace is shown here. The actual iSpace is in Japan, the virtual projection (on this picture) takes place in Hungary. We controlled the robot from the virtual environment in Hungary, and displayed the actual robot position as measured in Japan by the positioning system there.

5 Prediction of Mobile Robot Position

The network connection between the two laboratories has an average round-trip latency of 300 milliseconds. In order to compensate for the delay, the virtual robot's position must be predicted based on the parameters of its motion. Without this compensation, the network delay would result in an incoherent and fuzzy image of the virtual robot in Hungary.

Before this compensation takes place, we needed smooth positional data from the robot's positioning system in order to be able to predict its position. As stated earlier, the use of EKF on ultrasonic measurement and wheel encoder data reduced the measurement noise and thus the positional error. Our implementation of the EKF is described in detail in [15]. The combination of different sensor data (ultrasonic positioning system, laser-range finders and the robot's wheel encoders) is required in order to reduce position measurement errors. Raw sensor data is collected and processed (applying EKF) by iSpace components and is sent to the main computer, which handles the communication over the internet.

Network latency resulted in unacceptable visualization and inadequate control. This was mainly due to the uncertainty in the time delay, which was considerably high because of the wireless subnet in the real robot.

The following state equations were introduced to describe the position and the orientation of the robot:

$$\begin{bmatrix} \dot{x} \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \omega \end{bmatrix} \tag{1}$$

where x and y are the 2D coordinates and θ is the position-angle of the robot at any given time in the coordinate system that is tied to the floor of the room. Variable v denotes the speed of the robot and ω is the angular velocity around the center of the robot. Solving (1) in small discrete time intervals can predict the position of the robot, provided that the change in speed and angular velocity is not abrupt. In our case this latter condition will hold, since we will send commands in discrete time intervals to the robot over the internet, and predict the position of the local (virtual) robot during these short time intervals when speed and angular velocity is not changing.

The discrete form of (1) can be integrated with the above assumption. During time-step Δt both the speed and the angular velocity remains constant, therefore the position-angle for τ (that is in $[0, \Delta t]$) can be calculated as:

$$\theta(t+\tau) = \theta(t) + \omega(t)\tau = \theta(t) + \omega\tau \tag{2}$$

Now, integrating (1) is simple:

$$\begin{aligned} x(t+\Delta t) &= x(t) + \int_{\tau=0}^{\Delta t} v\cos(\theta(t+\tau))d\tau = x(t) + \int_{\tau=0}^{\Delta t} v\cos(\theta(t)+\omega\tau)d\tau = \\ &= x(t) + \frac{v}{\omega}(\sin(\theta(t)+\omega\Delta t) - \sin(\theta(t))) \end{aligned} \tag{3}$$

A similar solution can be written for $y(t+\Delta t)$, and we arrive to:

$$\begin{bmatrix} x(t+\Delta t) \\ y(t+\Delta t) \\ \theta(t+\Delta t) \end{bmatrix} = \begin{bmatrix} x(t) + \frac{v}{\omega}(\sin(\theta(t)+\omega\Delta t) - \sin(\theta(t))) \\ y(t) + \frac{v}{\omega}(\cos(\theta(t)+\omega\Delta t) - \cos(\theta(t))) \\ \theta(t) + \omega\Delta t \end{bmatrix} \tag{4}$$

Based on (4), a position predictor can be constructed, which predicts the robot's real response to the given commands.

The simulator uses the last synchronisation message $(x_m, y_m, \theta_m, T_m)$ where T_m is the time of measurement and the issued commands after $T_m : \{(v_c^i, \omega_c^i, t_c^i)\}$ which are ordered by the estimated time of execution t_c^0. For the first command is set to be equal to T_m. The state of the predictor is given by $(x_s^0, y_s^0, \theta_s^0, T_s^0 = T_m)$. During the estimation the predictor calculates the effects of the commands:

$$\begin{bmatrix} x_s^i \\ y_s^i \\ \theta_s^i \end{bmatrix} = \begin{bmatrix} x_s^{i-1} \\ y_s^{i-1} \\ \theta_s^{i-1} \end{bmatrix} + \begin{bmatrix} \frac{v_c^{i-1}}{\omega_c^{i-1}}(\sin(\theta_s^{i-1} + \omega_c^{i-1}(t_c^i - t_c^{i-1})) - \sin(\theta_s^{i-1})) \\ \frac{v_c^{i-1}}{\omega_c^{i-1}}(\cos(\theta_s^{i-1}) - \cos(\theta_s^{i-1} + \omega_c^{i-1}(t_c^i - t_c^{i-1}))) \\ \omega_c^{i-1}(t_c^i - t_c^{i-1}) \end{bmatrix} \quad (5)$$

The estimation of the position at time t is calculated from (5). Let k be the largest index for which $t_c^k < t$ holds.

The estimation is the following:

$$\begin{bmatrix} x^* \\ y^* \\ \theta^* \end{bmatrix} = \begin{bmatrix} x_s^k \\ y_s^k \\ \theta_s^k \end{bmatrix} + \begin{bmatrix} \frac{v_c^k}{\omega_c^k}(\sin(\theta_s^k + \omega_c^k(t - t_c^k)) - \sin(\theta_s^k)) \\ \frac{v_c^k}{\omega_c^k}(\cos(\theta_s^k) - \cos(\theta_s^k + \omega_c^k(t - t_c^k))) \\ \omega_c^k(t - t_c^k) \end{bmatrix} \quad (6)$$

Upon receiving a new synchronisation message the simulator updates the state by calculating the estimation for t_m and calculating a weighted average:

$$\begin{bmatrix} x_s^0 \\ y_s^0 \\ \theta_s^0 \end{bmatrix} = \alpha \begin{bmatrix} x^* \\ y^* \\ \theta^* \end{bmatrix} + (1-\alpha) \begin{bmatrix} x_m \\ y_m \\ \theta_m \end{bmatrix} \quad (7)$$

In current implementation this weight is set to $\alpha = 0.8$.

The commands, which are no longer affecting the prediction, are removed from the list and the estimated affection time of the first command is modified to be equal with the time of the synchronisation.

The proposed concept is summarized in Fig. 6.

Two different type of path have been executed in order to verify the behaviour of the predictor. In the following figures the real and virtual path are compared.

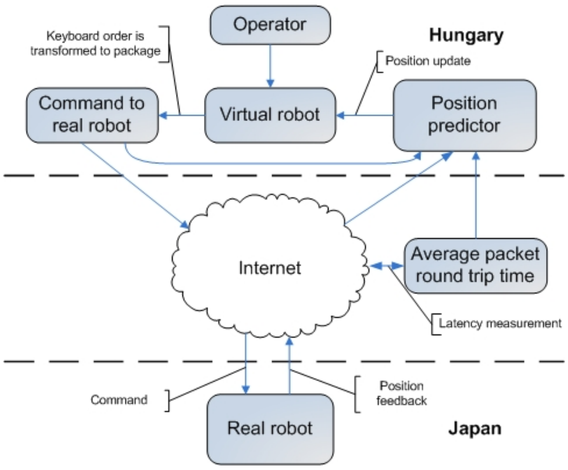

Figure 6
Concept of predictor

The first test case is shown in Fig. 7. This demonstrates the mobile robot following a variable curve. Based on the achieved results it can be clearly seen, that the virtual robot reacts faster to commands, than the real robot. This is caused by the fact, that the applied predictor does not take in account the acceleration model of the real robot and also because of the prediction of time delay of robot command execution. However the overall mean deviation is lower than 74.1 mm, which is an acceptable error rate in case of a very fast and simple predictor.

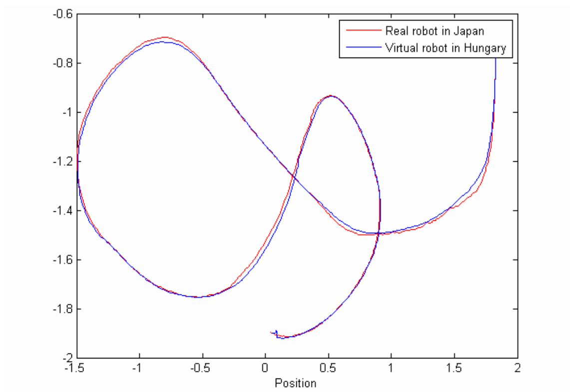

Figure 7
Test for variable curve following of mobile robot

In the second test case the robot was driven to predefined positions (-1, -0.9) and (-1.2, -1.7) and (0.1, -0.6) and (0.6, -0.8) and (1, -1.4), where the first coordinate is the x and the second y is respectively. The outcome of the test case can be seen in Fig. 8.

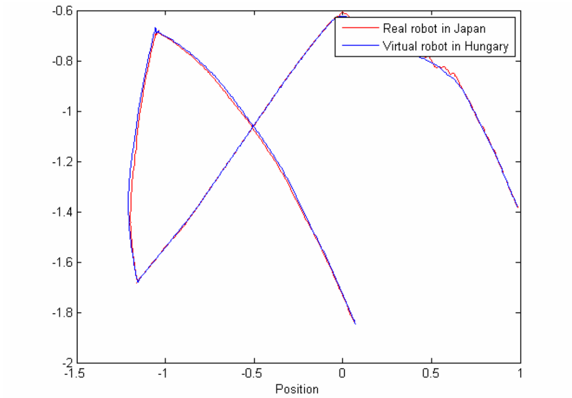

Figure 8

Test for destination approach of mobile robot

It can be observed, that in this case the predictor works better which is the result of the mainly linear motion. The mean deviation in this case is lower than 44 mm.

The current system setup only considers delays in one direction; the robot is visible at the estimated current position in real-time, but the control is still delayed according to the image and the one way latency.

Conclusions

In this paper the concept of extended Intelligent Space, stereoscopic virtual 3D copy, was presented. A mobile robot was guided simultaneously to desired locations in the real and remote space in our experiment. It can be divided to two main tasks: sensing and synchronization of robot navigation in the real and remote space. Sensing consists of tracking the position of the robot in the space. The synchronization is based on motion prediction and we have shown that it is an attractive and easy solution for this task. The presented implementation is an example of how to introduce physical agents inside Intelligent Space by utilizing the distributed sensors present in the space. It gives a practical solution that uses easily available sensors and robust control algorithms.

References

[1] P. Korondi, and H. Hashimoto, ”Intelligent space, as an integrated intelligent system”, Proc. of the International Conference on Electrical Drives and Power Electronics, pp. 24-31. 2003

[2] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale and S. Shafer, "Multi-Camera Multi-Person Tracking for EasyLiving", Proceedings of the 3rd IEEE International Workshop on Visual Surveillance(VS), pp. 3-10, 2000

[3] L. Rudolph, "Project Oxygen: Pervasive, Human-Centric Computing - An Initial Experience", Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAiSE), 2001

[4] B. Johanson, A. Fox and T. Winograd, "The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms", in IEEE Pervasive Computing 1/2, pp. 67-74, 2002

[5] T. Hori, A. Kaneko, K. Nagashima, Y. Nishida, H. Aizawa and H. Mizoguchi, "Delivery Task by a Humanoid Robot in the Sensorized Environment" in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 2930-2935, 2004

[6] J. H. Kim, Y. D. Kim and K. H. Lee, "The Third Generation of Robotics: Ubiquitous Robot", 2nd International Conference on Autonomous Robots and Agents, pp. 1-7, 2004

[7] B. K. Kim, N. Tomokuni, K. Ohara, T. Tanikawa, K. Ohba and S. Hirai, "Ubiquitous Localization and Mapping for Robots with Ambient Intelligence", in Proceedings of the IEEE international Conference on Intelligent Robots and Systems (IROS), pp. 4809-4814, 2006

[8] G. Sziebig, A. Gaudia, P. Korondi, N. Ando, and B. Solvang, "Robot vision for rt-middleware framework," in Proc. IEEE Instrumentation and Measurement Technology Conference (IMTC'07), Warsaw, Poland, pp. 1-6, 2007

[9] D. Brscic, "Mobile robot as physical agent of intelligent space", Journal of Computing and Information Technology, 2008, submitted for publication

[10] J. H. Lee, H. Hashimoto, "Intelligent Space - Concept and Contents", Advanced Robotics, 16/3 (2002), 265-280

[11] C. Cruz-Neira, D. Sandin, T. DeFanti, R. Kenyon, and J. Hart, "The CAVE: Audio visual experience automatic virtual environment". Communications of the ACM 35 (6), pp. 65-72, 1992

[12] ShapeWrap III Plus motion capture suit, Measurand Inc., http://www.motion-capture-system.com/

[13] Object-Oriented Graphics Rendering Engine, OGRE 3D, http://www.ogre3d.org

[14] OpenGL handbook: http://www.opengl.org

[15] D. Brscic, T. Sasaki, and H. Hashimoto, "Implementation of Mobile Robot Control in Intelligent Space", SICE-ICASTE International Joint Conference, 2006