NASA Conference Publication 10163

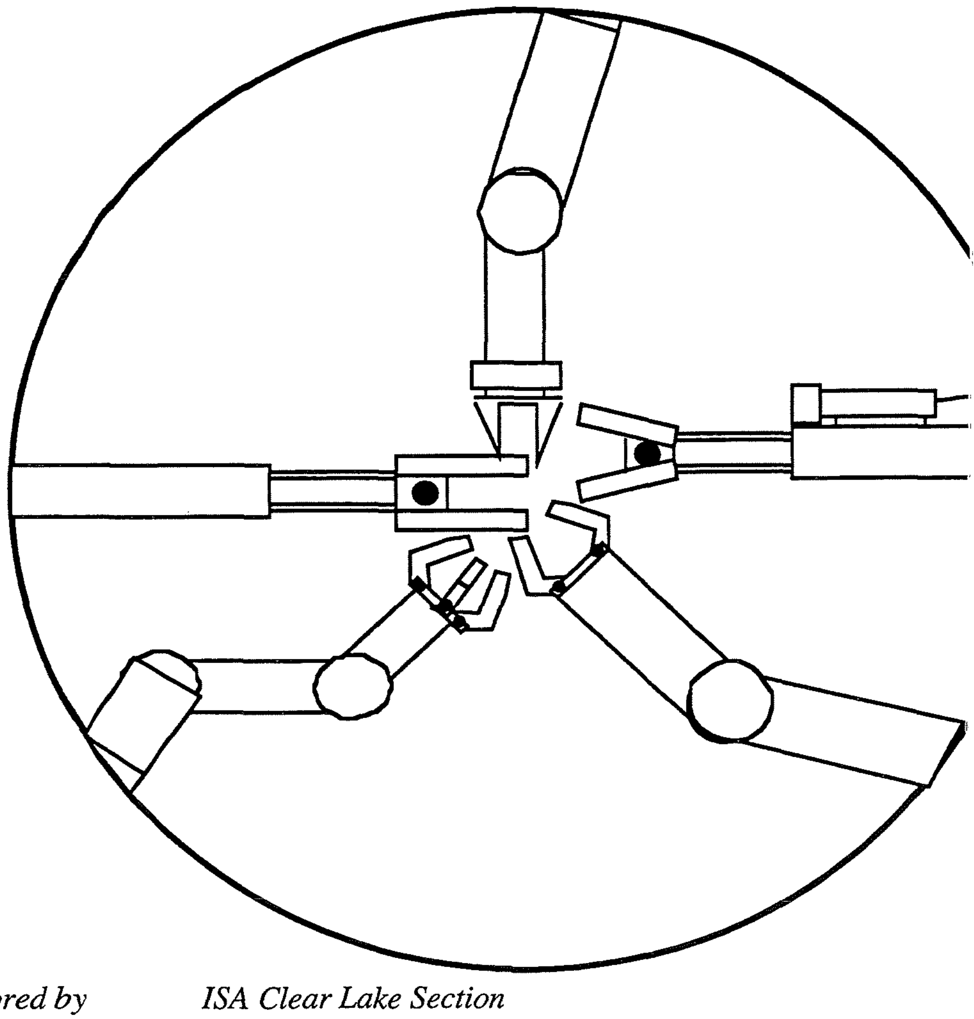
November 1994

ISMCR '94: Topical Workshop on Virtual Reality

Proceedings of the Fourth International Symposium on Measurement and Control in Robotics

Houston, Texas, USA

November 30 - December 3, 1994



Sponsored by *ISA Clear Lake Section*

Cosponsored by *IEEE Galveston Bay Section*

AIAA Houston Section

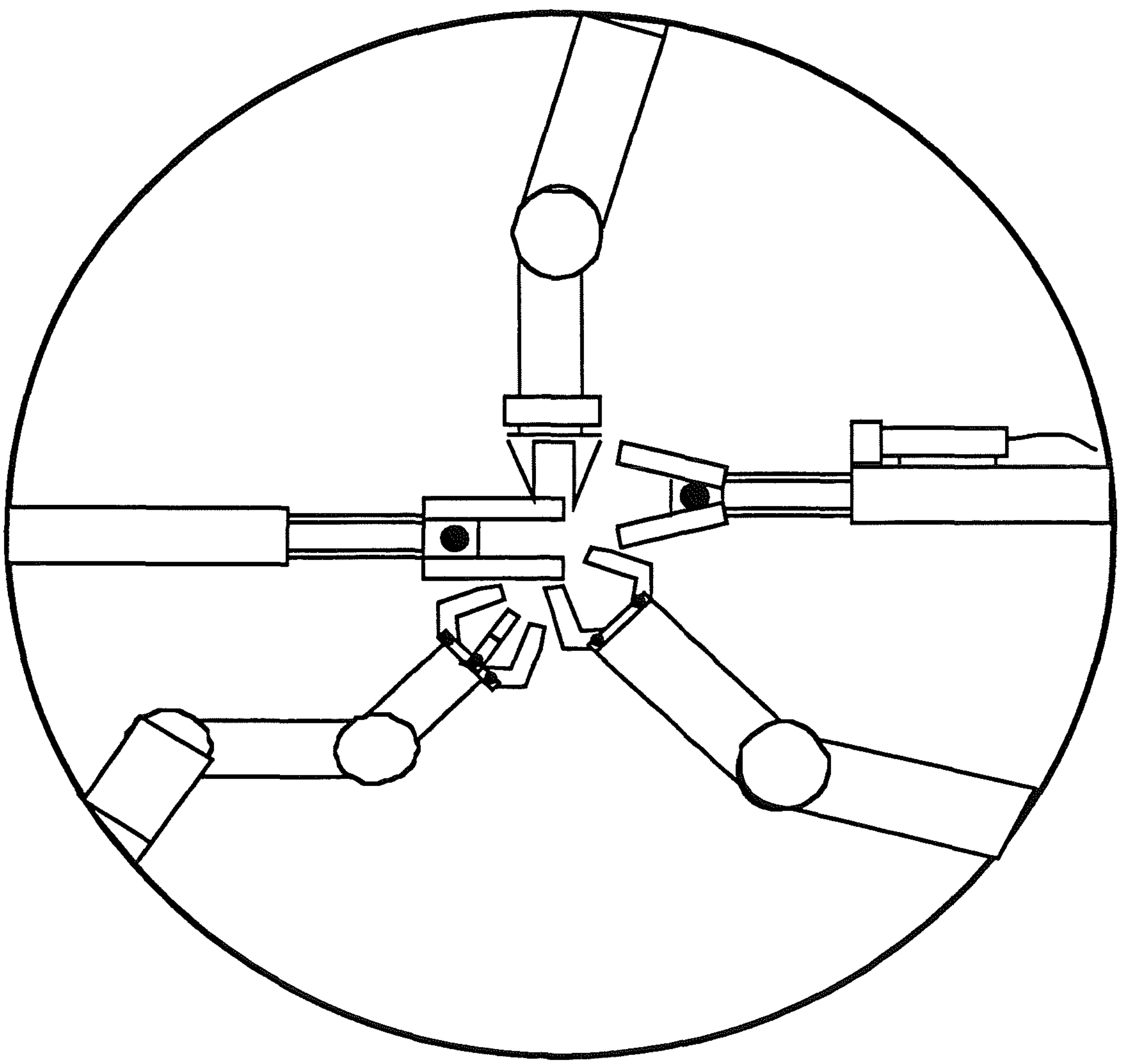
NASA Conference Publication 10163 November 1994

ISMCR '94: Topical Workshop on Virtual Reality

Proceedings of the Fourth International Symposium on Measurement and Control in Robotics

Houston, Texas, USA

November 30 - December 3, 1994



Organized by *IMEKO Technical Committee 17*
Hosted by *NASA JSC Automation, Robotics, and Simulation Division*
Clear Lake Council of Technical Societies

Library of Congress #94-69310
ISBN #0-9627116-1-6

This publication is available from the NASA Center for AeroSpace Information,
800 Elkridge Landing Road, Linthicum Heights, MD 21090-2934 (301) 621-0390

IMEKO Technical Committee on Robotics (TC-17)

Chairman
Prof. Susumu Tachi, University of Tokyo, Japan

Scientific Secretary
Dr. Zafar Taqvi, Dynacs Engineering, Inc., USA

Regional Program Coordinators
Dr. Daniel Biman, Slovak Technical University, Slovakia
Dr. K. K. Bhutani, Allahabad University, India
Vladimir Chudy, Slovak Technical University, Slovakia
Dr. Carlo Ferrero, Instituto di Metrologia "G. Colonnetti," Italy
Toshio Fukuda, Nagoya University, Japan
Prof. Duncan B. Gilmore, University of Queensland, Australia
Prof. K. T. Grattan, The City University, England
Prof. Patrick Kool, Vrije Universiteit Brussel, Belgium
F. Torres Leza, ETS Ingenieros Industriales, Spain
Prof. Andrzej Maslowski, Automotive Industry Research Institute
T. Pfeifer, RWTH Aachen, Germany
Charles Price, NASA Johnson Space Center, USA
J. Schanda, CIE Central Bureau, Austria
Reijo Tuokko, Tampere University of Technology, Finland
V. D. Zotov, Institute of Control Sciences, Russia
Qing Zhong Li, National Institute of Metrology, P. R. China

International Members
R.-J. Ahlers, Fraunhofer Institus fur Prozessautomatisierung, Germany
Kim Sun-Ho, Inst. Machinery & Metals, Korea
L. Kuhn, Czech Republic
Qing Zhong Li, National Institute of Metrology, P. R. China
Richard A. Mitchell, NIST, USA
A. A Petrov, Institute of Control Science, Russia
Bruno I. Rebaglia, Instituto di Metrologia "G. Colonnetti," Italy
Dusan Simsik, Technical College, Czech Republic
F. Torres Leza, ETS Ingenieros Industriales, Spain
Quangyou Xu, Tsinghua University, P. R. China

Workshop Committee

General Chairpersons	Susumi Tachi, University of Tokyo, Japan Norman H. Chaffee, NASA Johnson Space Center, USA
Program Chairpersons	Robert H. Brown, Comprehensive Solutions, USA David Homan, NASA Johnson Space Center, USA
Host Chairperson	Andrew Lindberg, Clear Lake Council of Technical Societies, USA
General Coordinator	Zafar Taqvi, Dynacs Engineering, USA
Publications Chairperson	Jon D. Erickson, NASA Johnson Space Center, USA
Registration Chairperson	Mary Stewart, HEI, USA
Finance Chairperson	Ernie Hillje, Rockwell International, USA

Contents

Technical Paper *Page*

Session 1: Rendering
Dr. Stephen R. Ellis, Chair
Masatoshi Ishikawa, Co-Chair

Session 2: Applications
Prof. Andrzej Maslowski, Chair
David Homan, Co-Chair

Session 3: Tracking Sensors
Prof. Hideki Hashimoto, Chair
Dr. Jon Erickson, Co-Chair

Contents
(continued)

Technical Paper *Page*

Contents
(continued)

Technical Paper *Page*

Session 6: Applications
Prof. Steve Benford, Chair
Dr. Jayant Ramakrishnan, Co-Chair

Session 1
Rendering

VIBRATORY TACTILE DISPLAY FOR TEXTURES

Yasushi IKEI, Akihisa IKENO and Shuichi FUKUDA

Tokyo Metropolitan Institute of Technology
6-6 Asahigaoka, Hino-shi, Tokyo 191, Japan

Abstract:
We have developed a tactile display that produces vibratory stimulus to a fingertip in contact with a vibrating tactor matrix. The display depicts tactile surface textures while the user is exploring a virtual object surface. A piezoelectric actuator drives the individual tactor in accordance with both the finger movement and the surface texture being traced. Spatiotemporal display control schemes were examined for presenting the fundamental surface texture elements. The temporal duration of vibratory stimulus was experimentally optimized to simulate the adaptation process of cutaneous sensation. The selected duration time for presenting a single line edge agreed with the time threshold of tactile sensation. Then spatial stimulus disposition schemes were discussed for representation of other edge shapes. As an alternative means not relying on amplitude control, a method of augmented duration at the edge was investigated. Spatial resolution of the display was measured for the lines presented both in perpendicular and parallel to a finger axis. Discrimination of texture density was also measured on random dot textures.

Keyword: Tactile Display, Vibrotactile Sensation, Surface Texture, Duration Time, Spatial Resolution, Density Discrimination, Virtual Reality

1. INTRODUCTION

Force reflection devices to somatic sensation have been developed in various configuration designs for the purpose of teleoperation from the 1960s. Recently emerging needs of such devices from virtual reality technology are again accelerating the research regarding these haptic feedback devices. When the user of such system interacts with physical objects presented virtually, however, force reflection device alone is insufficient. Tactile sensation, by which the shape and surface texture are perceived, is also crucial to increase the sense of presence of displayed object. In addition to the deep sensation presented by force feedback devices, the cutaneous sensation plays an important role particularly in ensuring the sensor modality of a human operator, by which cognitive cues are diversely provided in ordinary environment.

The surface texture sensation depends on many aspects of physical properties specifying the object surface, such as microscopic geometry, friction coefficient, kinetic elasticity, thermal conductivity, etc. Modern study on such tactile texture perception was originated by Katz(1925) [1] who set many agenda on the subject. Recently, Hollins(1993) [2] proposed a perceptual space in which surface texture properties were discriminated within a three-dimensional model. To present a virtual texture in the tactile space, the physical properties of a surface should be well-imitated in above senses, however, it is extremely difficult to reproduce all of these properties. Therefore, an effective scheme, by which the tactile sensation is purposively stimulated, has been an open interest in the research area [3]. One solution to that difficulty is to reduce contact dimension to a single point that explores within the textured surface. Such devices were developed by Minsky(1990) [4] and by Akamatsu (1994) [5] that produced surface texture sensation as traced by a point not representing it to the finger as the two-dimensional surface to be sensed simultaneously. This approach contributes to device simplicity, however a part of intrinsic properties of spatially distributed cutaneous sensation is dismissed unused.

A vibratory stimulus, produced by a mechanical device, has been investigated as an effective instrumentality to transmit information to the blind from the 1960s [6]-[8]. The Optacon is the typical device that employs vibratory stimulus to convert optical information to tactile sensation [9]-[10]. The device was first developed by Linvill(1966), and commercially

This research was supported by the special research fund of TMIT.

available from 1971. Several display control modes were tested to represent letters on the Optacon by Craig(1981) [11]. Since the principal purpose of the Optacon was reading aid for the blind, the method for texture replication had not been treated as a principal theme.

We have developed a vibratory tactile display for presenting sensations related to the texture on object surfaces. This device has vibrating tactors similar to the Optacon, however the individual tactor can be controlled with much extended flexibility in spatiotemporal pattern generation. Basic characteristics of the presented sensation were investigated experimentally, in which control schemes were discussed in terms of representing simple edged and random dot textures.

2. MECHANISM OF VIBRATORY TACTILE DISPLAY

A prototype system of the vibratory tactile display is shown in Figure 1. Vibratory stimulus is given to the index fingertip pad placed on a display window, 10 x 20 mm^2, at the top of the display box. The display window is a matrix of 'tactor,' a display element made of a piano wire 0.5 mm in diameter. Within a matrix, 5 x 10 tactors are disposed with a 2 mm pitch forming a rectangular window (Figure 2). Each tactor is driven at 250 Hz by a piezoelectric actuator attached to a magnifying mechanism to yield about 80 micron amplitude. This frequency of vibration was adopted for its highest sensitivity on the basis of the equal sensation magnitude curve measured by Verrillo [6].

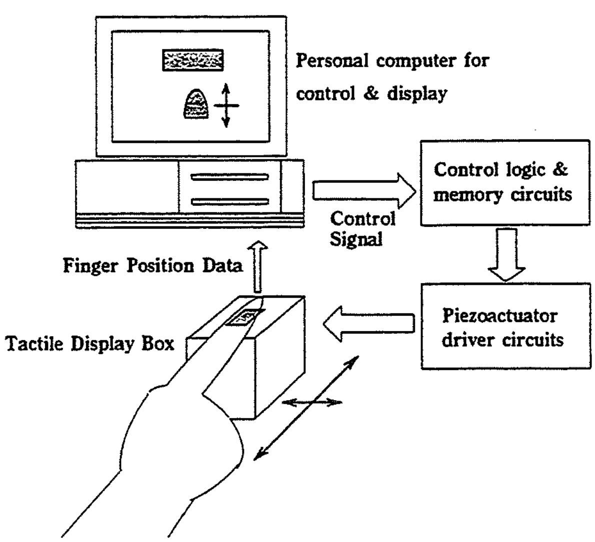


Figure 1 Tactile Display System

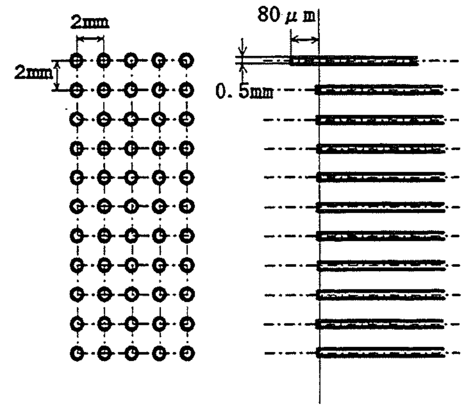


Figure 2 Five-by-ten matrix of tactor with 2mm pitch.

The user of the display explores a surface of a virtual object with the fingertip fixed on the window, moving his/her hand within a two-dimensional plane together with the display box. Position data of user's finger, compatible with the position of display, is tracked by a mouse attached to the display box. On the basis of the position data, which is equivalent to relative finger movement within a presented texture, a personal computer controls spatiotemporal patterns of tactor vibration. In addition, the computer also carries out the renderings on a CRT display showing CG images of both the finger and virtual textures.

3. STIMULUS GENERATION SCHEMES

The term, surface texture, in tactile sense is used here as a geometrical profile of an uneven plane that contains little difference of levels or protrusions that make inherent tactile stimuli when it is traced by a fingertip. Other properties of surface that are to contribute to tactile sensation such as frictional, kinetic and thermal characteristics are not considered to represent, although they cannot be eliminated completely from our physical embodiment of the display. Thus for the first step, we treat the texture as a binary valued two-dimensional plane; the plane has 'high' and 'low' portions extended similarly to a binary picture image. Given the simplified surface, a basic and natural mapping from a texture to the display window, by which each tactor is driven to make stimulus, is that tactors in high portions in the texture generate vibratory stimulus in the display window. We incorporate this fundamental mapping in the display control as its static phase. The dynamic mapping to realize temporal property of tactile sensation has much alternatives to be discussed as shown in the following.

3.1 Stimulus duration for reproduction of adaptation process

Tactile sensation of object surface texture is usually obtained while we explore the surface with the fingertip; minute protrudent profiles of the texture afford us two-dimensionally varying stimulus during the exploration. In doing that to examine the surface, we incidentally stop the finger movement and restart it again not always intentionally, where the process of sensor adaptation occurs. After the movement stopped, the sensation of touching a surface gradually decreases. The decay time of the adaptation differs depending on the kinds of mechanoreceptor ranging from a few milliseconds to tens of seconds. The decay time of vibration receptors is very short, a few milliseconds, and that of touch receptors has a range from about fifty to five-hundred milliseconds [14].

Without taking account of this sensory adaptation, the display does not give a good representation of tactual texture. Since, if the display continuously activates the high portion of texture after the finger stopped, the stimulus will be too intense for static touching, and it will cause unwanted paralysis of cutaneous sensation. Moreover, if it terminates the stimulus simultaneously with a finger stop, the impression of the surface texture becomes very queer as though the texture touching suddenly vanished.

To incorporate this adaptation process in stimulus producing control of the tactile display, the generated vibration must be adjusted temporally with respect to its intensity. However, the amplitude of individual tactor vibration cannot be regulated on this display, because the analog circuits to alter driving voltages would be too large to implement. Consequently, we have selected a method to give an appropriate duration to each tactor after the finger exploration stopped. A proper duration of vibration equivalent to diminishing sensation was able to simulate the adaptation process to a good extent.

Experiment

The effective duration time was measured by the method of adjustment where a single line edge was presented on the display perpendicularly to a finger axis. In the experiment, a standard stimulus of a fixed fine wire 0.5 mm in diameter was provided aside the display at the same height as the display plane. On the display while the display box has a velocity, just a single row of tactors was excited to avoid the termination of vibratory stimulus. (This assumes the virtual edge width to be 2 mm, however it introduced a little uncertainty of edge position rather than its extended width.) Subjects were instructed to trace the standard wire by the right index finger for about five times before each testing session. Then the subject rested the index finger horizontally on the display window holding the display box by other fingers. A vibratory stimulus was then generated on the display window while the subject was tracing on a virtual wire. During the experiment, the subject wore headphones through which a white noise was presented to reduce auditory cues and distraction due to the sound of the display. After the finger movement stopped, the stimulus was extended by the initial duration time randomly set each time within either the ranges [0, 10] or [40, 50] milliseconds for ascending and descending series, respectively. The subject was allowed to change the duration by the adjustment keys of ±1, ±3, and ±6 msec allocated on a keyboard, until he judged the similarity between the displayed wire and the real one was maximized. Each experiment for a subject consisted of twenty trials, ten for both series. The number of subjects was four, including a female, in their 20s or 30s; two were inexperienced with only a few rehearsal before the experiment.

Results

Figure 3 shows adjusted duration mean times of four subjects for each series of adjustment directions. An analysis of variance reveals significance at the .01 level in all of subject differences ($F=20.9$, $df=3/72$), series effects ($F=98.4$, $df=1/72$), and the subjects by series interaction ($F=14.8$, $df=3/72$). Subject B and C were inexperienced, and they exhibited large mean time differences between ascending and descending series. It seems that they had not precisely perceived the results of their own changes in the case of

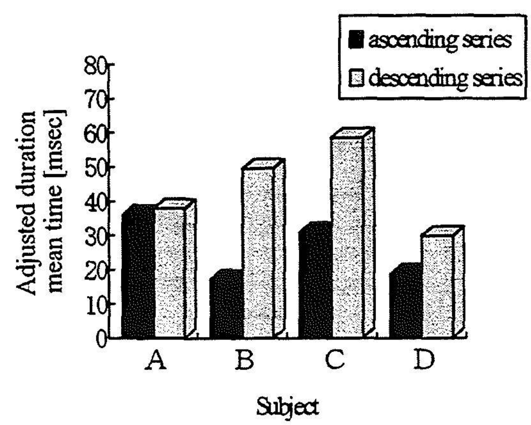


Figure 3 Adjusted duration time for a virtual wire

descending series, as they sometimes increased the duration time moreover. However in ascending series, adjusted mean times were not so small that they should be judged not realizing the duration at all. Consequently it appears that a duration no less than about ten milliseconds has an effect to produce an after image gradually vanishing at the user's finger. A possible duration time to implement in the display may be an average time of ascending series of both experienced and inexperienced users. Then, the time obtained in this experiment was 25.6 msec. In general, subjective impression was fairly good when the duration was properly added. A support of this figure of duration time may be obtained from a fact that the time threshold of tactile sensation is said to be around it, for example, about 27 millisecond, that agrees with our experimental data.

3.2 Representation methods of other simple edges

For implementing fundamental displayed elements to represent the rugged surface texture, we examined other simple edge patterns, illustrated in Figure 4, that included protrusions and retractions, or recesses, wider than a single wire. With regard to the protrusion such as Figure 4(a) of width over two millimeters, it was not appropriate to assign simply vibrating tactors to protrudent regions, since the edges at the region boundary were blurred consistently with the increase of the width. The sensed image at the edge was observed as rather a gentle slope than a definite line.

An alternative assignment of vibrating tactors to avoid the diffusion of edge image was examined, that limited the vibration tactors only at the edges where tactors inside the edges being suppressed. However, the method was not effective to represent the protrudent shape since it produced hollow impression at the inside between the edges; the image was rather near to Figure 4(b) when the finger was *on* the protrusion.

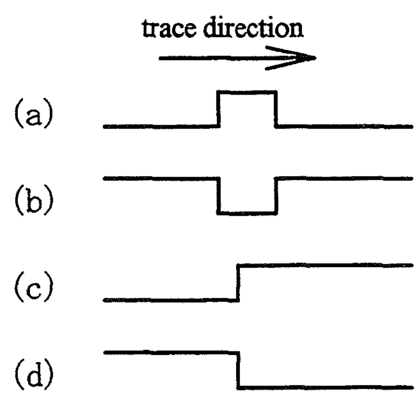


Figure 4 Fundamental texture elements of line edge.

The similar discussion was valid in the cases of Figures 4(c) and 4(d). Eventually, if all tactors within the high region were excited, the impression given from the region was rather a gentle sloped swelling attended with some sense of friction than a plate with a sharply defined edge. Moreover, if the tactors at the edge alone were excited, a high plate region was not perceived but a single edge was observed, naturally.

From above discussion, it is found that stimulus distribution control is required where the stimulus density, or intensity, varies across the edge, incorporating the directions of low to high or high to low. However, the vibration intensity is basically constant in this device at present. Thus, making a spatial or temporal intensity gradient should be an alternative scheme; a temporal modulation to alter intensity was examined tentatively.

The basic vibration at 250 Hz was ceased from 10 to 90 percent at 25 Hz modulation interval to obtain intensity gradient. The result of this scheme, however, was not necessarily effective, since additional frequency spectrum of 25 Hz caused another quite different sensation that surpassed the decrease of basic frequency stimulus.

3.3 Augmented duration method for edge representation

Another method to represent the edge is to utilize duration difference after the finger stopped. Since the edge is localized especially when the finger stopped on the edge, a longer duration than that set inside the protruded area can easily highlight the edge. Preferable duration time was measured by the method of adjustment.

Experiment

A 20 x 20 mm virtual plate was assumed and displayed on a CRT as a square region. Subjects traced the virtual plate one dimensionally, back and forth in parallel to the finger axis, stopping on both edges, onto the plate and off from the plate. After the finger exploration movement stopped, all tactors *on the plate* were extended vibration by the same duration time of 10 msec; this duration time was around the minimum that could avoid inadequate vanishing image and ensure duration contrast to the edges. The tactors on the rising edge, that had just climbed onto the plate, were assigned 30 msec of constant duration time; which was about the mean time selected previous experiment. This side edge did not require additional duration other than a normal duration, since the finger region that had not reached the plate was free from stimulus which did produce high contrast of stimulus at the boundary. On the other side falling edge, across which the finger had just relieved a vibration stimulus, a longer duration was required to emphasize the edge to the finger which had degraded sensitivity after experienced vibration area of the virtual plate.

Before each session of twenty trials, ten for both ascending and descending series, subjects were required to trace a standard plastic plate. The initial duration time was set randomly in either the ranges of [0, 20] or [200, 220] milliseconds for ascending and descending series, respectively. The adjustment keys allowed to change duration were ±3, ±10, and ±30. Two experienced male subjects in their 20s and 30s performed the experiment with the masking headphones.

Results

Figure 5 shows the duration mean times in which the subject difference was not significant; a series effect was significant at .01 level ($F=17.5$, $df=1/36$); the subjects by series interaction was significant at .05 level ($F=5.0$, $df=1/36$). Means of ascending and descending series were 125.5 ms and 95.0, respectively. Both series had crossed mean values in the adjustment from initial values that may be attributed to the error of habituation. A tentative standard duration time for a falling edge seems to be the series mean of 115 ms.

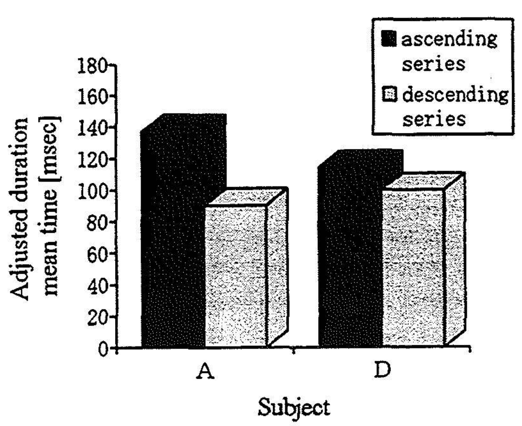


Figure 5 Adjusted duration time for a falling edge

3.4 Spatial resolution

A spatial resolution is a common index frequently referred to in describing the performance of a display device. One-dimensional resolution of the tactile display was examined in both the horizontal and vertical directions. While the resolution specified by the number of lines to be counted does not directly describe overall presentation power with regard to tactually perceivable surface texture, it seems to yield suggestive information by which the relation between this device and tactile sensation is extensively discussed.

Method

Several lines of virtual protrusions were displayed both perpendicular and parallel to the finger axis within the test region 60 mm in length, where a visual image of protrusions was suppressed and only a boundary frame was displayed. The line pitch was changed in thirteen cases as shown in Table I, and the ratio of protrusion width to a pitch was altered in three cases of 25, 50, and 75 % as illustrated in Figure 6 of perpendicular allocation. Subjects were asked to report the number of lines. The experiment was repeated ten times for each pitch, randomly selecting a pitch from the pitch set. The data was obtained from three male subjects in their twenties as the previous experiment.

Table I Line pitches selected in thirteen ways.

pitches of line (mm)						
0.8	1.2	1.6	2.0	2.4	2.8	4.0
6.0	8.0	12.0	16.0	20.0	24.0	

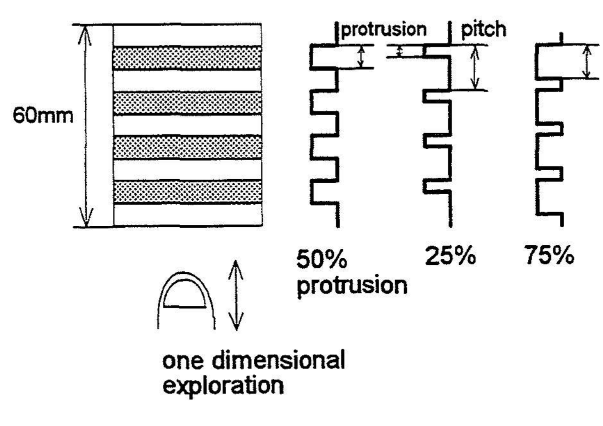


Figure 6 One dimensional virtual edges perpendicular to the finger axis

Results

The results are shown in Figures 7 and 8 for perpendicular and parallel cases, respectively. The ordinate of Figures is the ratio of correct answer, and the abscissa is a pitch of lines. In the correct answer ratio, the response within plus and minus twenty percent errors are included, since counting lines by fingertip exploration is rather difficult to contain an inevitable miscount, even if it is done on a real surface with physically engraved lines. (Beforehand, we have conducted a preliminary experiment estimating the line counting ability of fingertip on real line-carved samples produced by a rapid photo forming machine. The samples were shaped to realize edge patterns illustrated in Figure 6, where the height of edges was 0.5 mm; the height actually had no significant effect on the discrimination of lines. Thereby, correct counting of lines was measured to require at least about 4 to 6 mm pitch.)

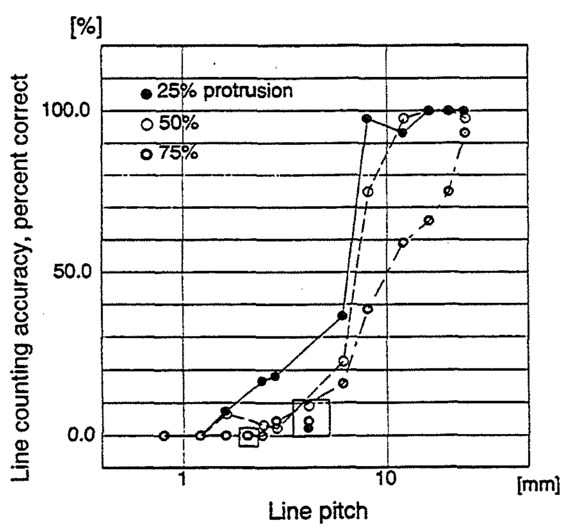


Figure 7 Line counting accuracy, where lines were presented *perpendicular* to the finger axis.

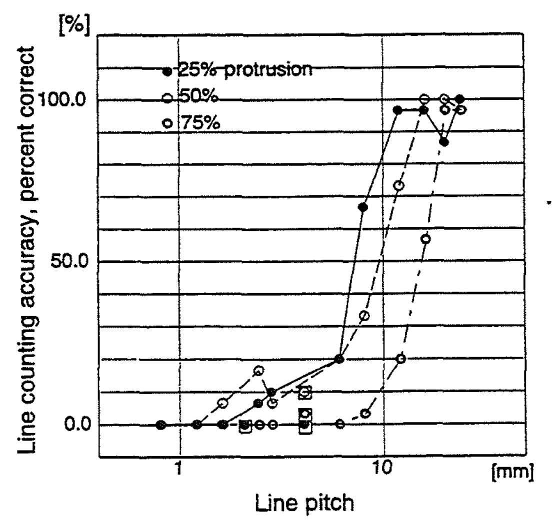


Figure 8 Line counting accuracy, where lines were presented *parallel* to the finger axis.

In the case of the display, the correct answer ratio reached over seventy-five percent at 8 mm pitch with the exception of 75 % protrusion case. This value seems acceptable taking account of the display's tactor pitch of 2 mm. In the case of 75 % protrusion ratio, it was more difficult to discriminate the 'low' position between lines than the other cases. Therefore, the counting accuracy was slow to rise.

The data points enclosed in a open square at 2 and 4 mm pitches are abnormal values, since the pitches are equivalent to multiples of display tactor pitch of 2 mm. Displaying the lines in these singular pitches produced a synchronized vibration of all tactors, where the vibration was far from the usual sensation experienced through tracing a physical surface.

Figure 8 indicates the result where the lines were presented parallel to the finger axis, and the finger exploration movement was valid only in the lateral direction. A general difference from the data obtained on perpendicular lines is lower counting accuracy at almost all line pitches; the decrease at eight millimeter pitch is remarkable. Twelve millimeter pitch was required for almost correct line counting at 25 % protrusion ratio in the parallel case, while eight millimeter pitch was sufficient for the lines displayed perpendicularly.

3.5 Discrimination of texture density

Natural surface textures in general produce multi-level stimulus magnitudes, that is equivalent to gray scale in a visual image, as well as sharp edges described in the previous section. To represent the multi-level textures, it is required first to determine the number of levels that can be displayed by this tactile display. Here it is assumed that multi-level textures are approximated by a binary dot image. The perceivable density difference of binary random dots was examined by presenting a pair of regions that had different mean densities.

Experiment

Some example textures that were used in the experiment are shown in Figures 9 - 11. The size of the texture area was 30 x 60 mm; each dot size was 2 x 2 mm. A black dot in the texture excited the display pin vibration. The texture area was divided vertically

(U:25% L:75%) (U:40% L:60%) (U:50% L:50%)

Figure 9 Sample textures of 50 % mean density.

(U:15% L:65%) (U:30% L:50%) (U:40% L:40%)

Figure10 Sample textures of 40 % mean density.

(U:10% L:50%) (U:20% L:40%) (U:30% L:30%)

Figure 11 Sample textures of 30 % mean density.

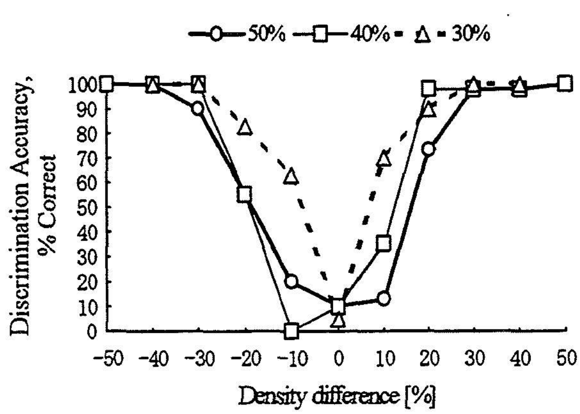


Figure 12 Discrimination accuracy vs. Density difference.

into two regions to provide density difference between the upper and lower sides by the percentages from -50 % to +50 %. In Figure 8, all of three textures have the same mean density of black dots of 50 %; in Figures 10 and 11, the mean densities are 40 % and 30 %, respectively. Subjects were required to judge which side was more dense after exploring both sides. On the visual display, the rectangular wire frame of the area and a cursor that indicated the finger position were presented. The texture with a particular difference was presented ten times in one session that included 110 trials for each of three mean density of 50, 40, and 30 %. Subjects wore the headphones similarly to previous experiments. Four subjects in 20s and 30s, including a female, executed this experiment.

Results

Figure 12 shows the ratio of correct answer as a function of density difference between the upper and lower sides. The ratio is the average data of four subjects. Plotted circles indicate the case where the mean density was 50 %, and squares for 40 %, triangles for 30 %, respectively. In cases of 50 and 40 % mean density textures, thirty percent difference in both sides gives almost complete discrimination, and about 20 % difference is a transition point. The lower mean density of 30 % exhibited improved accuracy of discrimination where 10 % difference was perceived much more than other two cases.

4. DISCUSSION

One of principal surface texture elements sensed by a fingertip is a line edge within a plane. We have referred to the possibility to employ duration for representing the edge shape. This is because we thought that to replicate the after image of vanishing sensation increases the similarity of touch impression to the real edge and it can be well simulated by the stimulus duration added properly after the finger movement stopped. Moreover, such stimulus control that terminates tactor vibration after the duration is an appropriate scheme which can avoid sensation paralysis. (Although the Optacon does not employ such termination of vibration, since its purpose is not rendering of surface texture image but translating symbolic letters of printed matter distinctively.)

A single edge and a boundary edge of a plete were presented fairly well by the augmented duration menthod. However, presenting the even inside of a plate bounded by edges was not suited to this display, since the display can not directly give a shearing force that must be introduced by finger tracing movement, although it does display a little sense of friction. The lack of shearing sensation is compensated in the edge representation case by an apparent movement that can be sensed as long as the surface has any variation in the geometrical state.

Another principal element of texture is a periodical variation of a surface. Repetition of lines is presented with ease by the display in the sense of perception, not counting. In the spatial resolution data obtained, the line counting under four millimeter pitch has been discussed equally to other pitches. However, tactile pattern recognized in this range of spatial frequency must be inaccurate taking account of the Nyquist criterion. Further analysis of the display spectrum is required from the tactile sensation point of view.

Spatial resolution as a counted value was not necessarily high enough, which is partially due to the display tactor pitch of 2mm. Regarding the spatial sensitivity of a fingertip, spatial resolution has been measured by Weinstein (1968) [12] and others, and referred to as ranging from 2 to 4 mm for a simultaneous spatial threshold, or two-point limen. Consequently in this sense, the display tactor pitch might be adequate in a sense. However this value of spatial threshold is valid while the stimuli is statically provided. The successive spatial threshold, where two stimuli are presented sequentially, is reported on the order of 10 to 30 times smaller than that (Loomis, 1978) [13]. Accordingly the presentation bandwidth of the display, in which the surface image is dynamically produced, is considered to be restricted by its tactor pitch. Display control needs more extended schemes for surmounting this hardware limitation.

Tactile discrimination ability of mean intensity of random dots was close to that of vision although whole texture patterns are not shown here due to space limitaiton. That was an unexpected result in contrast to the result of spatial resolution experiment. This leaves one possibility to present a gray scale image in this form. Future work to augment the display presenting capacity also includes intensity control which is invoked partially by hardware controlling frequency and phase. Tactile sensitivity to the frequency change and phase offset is acute according to our tentative observation. These parameters will surely contribute to rendering versatility of the display, especially in representation of gray scale images.

REFERENCES

1. Katz, D., "The World of Touch (L. E. Krueger, Trans.)," Hillsdale, NJ: Erlbaum, 1989. (Original work published 1925)
2. Hollins, M., Faldowski, R., Rao, S., and Young, F., "Perceptual dimensions of tactile surface texture: A multidimensional scaling analysis," Perception and Psychophysics, 54(6), 1993, pp. 697-705.
3. G. J. Monkman, "An Electrorheological Tactile Display," Presence, 1(2), 1992, pp. 219-228.
4. Minsky, M., Ouh-young, M., Steele, O., Brooks, F. P., and Behensky, M., "Feeling and Seeing: Issues in Force Display," Computer Graphics, Vol. 24, No. 4, ACM SIGGRAPH 1990, pp. 235-243.
5. Akamatsu, M., Sato, S. and MacKenzie, I. S., "Multimodal Mouse: A Mouse-Type Device with Tactile and Force Display," Presence, Vol. 3, No. 1, 1994, pp. 73-80.
6. Verrillo, R. T., Fraioli, A. J. and Smith, R. L., "Sensation magnitude of vibrotactile stimuli," Perception & Psychophysics, 6(6A), 1969, pp. 366-372.
7. R. T. Verrillo, "Subjective Magnitude Functions for Vibrotaction," IEEE Transaction on Man-Machine Systems, MMS-11(1), 1970.
8. J. H. Kirman, "Tactile apparent movement: The effects of interstimulus onset interval and stimulus duration," Perception and Psychophysics, 15(1), 1974, pp. 1-6.
9. J. G. Linvill and J. C. Bliss, "A direct translation reading aid for the blind," Proc. IEEE, 54, 1966, pp. 40-51.
10. J. C. Bliss, "Dynamic tactile displays in man-machine systems, " IEEE Transactions on Man-Machine Systems. (Special issue: Tactile displays conference), 11(1), March 1970.
11. J. C. Craig, "Tactile letter recognition: Pattern duration and modes of pattern genaration," Perception and Psychophysics, 30(6), 1981, pp. 540-546.
12. S. Weinstein, "Intensive and extensive aspects of tactile sensitivity as a function of body part, sex and laterality", in The skin senses, D. R. Kenshalo(Ed), Springfield, Ill: Thomas, 1968.
13. J. M. Loomis and C. C. Collins, "Sensitivity to shifts of a point stimulus: An instance of tactile hyperacuity," Perception and Psychophysics, 24(6), 1978, pp. 487-492.
14. Schmidt, R. F., Ed., "Fundamentals of Sensory Physiolosy," Springer-Verlag KG, Tokyo, 1986.

APPLIED VIRTUAL REALITY AT THE RESEARCH TRIANGLE INSTITUTE

R. Jorge Montoya
Senior Research Engineer
Manager, Virtual Reality Group
Research Triangle Institute
P.O. Box 12194
RTP, NC 27709-2194

ABSTRACT

Virtual Reality (VR) is a way for humans to use computers in visualizing, manipulating and interacting with large geometric databases. This paper describes RTI's VR infrastructure and its application to marketing, modeling, architectural walkthrough, and training problems. VR scientific integration techniques used in these applications are based on a uniform approach which promotes portability and reusability of developed modules. For each problem, a 3D object database is created using data captured either by hand or electronically. The objects' realism is enhanced through either procedural or photo textures. The virtual environment is created and populated with the database using software tools which also support interactions with and immersivity in the environment. These capabilities are augmented by other sensory channels such as voice recognition, 3D sound, and tracking. Four applications are presented: a Virtual Furniture Showroom, Virtual Reality Models of the North Carolina Global TransPark, a Walkthrough of the Dresden Frauenkirche, and the Maintenance Training Simulator for the National Guard. Degree of realism and update rate requirements for these applications posed significant implementation challenges which were met in every case. These applications demonstrate the viability of VR and show great promise for VR as a cost effective marketing, training, and teaching tool.

INTRODUCTION

Virtual Reality is an exciting new approach to human-computer interactions. Based on long-established computer graphics techniques and benefiting from recent advances in computer hardware and software, this technology supports the creation of and interaction with "worlds" which are either faithful replicas of existing ones or evoke the existence of yet-to-be-created ones. In its purest form, VR is the presentation of and interaction with a synthetic, computer-generated, 3D world, so realistic that the user feels as if he/she were experiencing reality. Over two and one-half years ago, RTI made a market and technology analysis that concluded that virtual reality was a technology poised for transfer from basic research laboratories to applied research institutions such as RTI. Based on this conclusion, RTI developed a business plan that defined the VR market segment which it would pursue, identified areas in which it would invest internal research and development (IR&D) funds, and mapped out the hardware and software configuration necessary for the development of an advanced VR laboratory which would serve as the foundation for its work in Virtual Reality.

The market analysis recommended that RTI concentrate in a market consisting of architectural walkthroughs, marketing, rapid prototyping, and training applications. These application areas would allow RTI to take advantage of its strong multidisciplinary background and provide value added as scientific and engineering integrators with the appropriate mixture of technology and domain experts to accomplish a specific job.

This paper describes the RTI VR Laboratory infrastructure and several VR projects which the institute has undertaken. The projects described in this paper are examples of the application of Virtual Reality to marketing (a Virtual Furniture Showroom), planning (Virtual Models of the North Carolina Global TransPark), architectural walkthrough (a Walkthrough of the Dresden Frauenkirche), and training (A Maintenance Training Simulator for the National Guard).

The paper includes a description of the methodology used to implement the various applications and a discussion of the system performance achieved in each of these applications.

THE RTI VIRTUAL REALITY LABORATORY

RTI is an independent, not-for-profit corporation founded in 1958 by the University of North Carolina at Chapel Hill, North Carolina State University, and Duke University. RTI conducts applied and basic multidisciplinary research for governmental agencies and commercial clients.

RTI has developed a nationally recognized program in computer graphics applications over the past twenty years [1, 2, and 3]. The emphasis of RTI's work in computer graphics has evolved and advanced as hardware capabilities have improved and software tools have become more sophisticated [4 and 5]. In keeping with its tradition of conducting advanced multidisciplinary applied research, RTI has established a state-of-the art Virtual Reality laboratory with an investment of well over $1,000,000 over the last two years.

Hardware:

The VR laboratory infrastructure is shown in Figure 1. The backbone of the laboratory is a network of computers that includes the full range of VR capabilities. The computing environment consists of platforms ranging from PCs (Pentiums, 486s, and Apple's Quadra 840 AVs) to the Silicon Graphics deskside Crimson Reality Engine and the rack mounted Onyx Reality Engine2. It also includes two IBM RS-6000s, a model 320 and a model 570. The environment also supports full immersion with a Virtual Research's EyeGen3 head mounted display (HMD) and see-

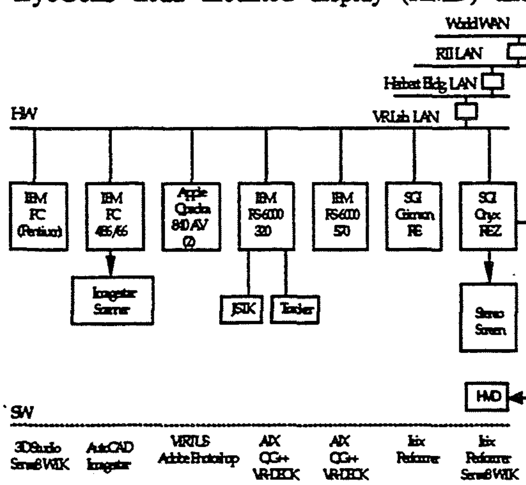


Figure 1 RTI VR Laboratory Infrastructure

through immersion with Stereographics' CrystalEyes shuttered glasses; tracking with a Polhemus magnetic tracker and an acoustic Logitech tracker; navigation with a joystick or mouse; a stereo projection capability using the VREX-1000 system; a wide range of modeling and rendering software environments; and speech recognition and sound output capability.

The Crimsom and the Onyx are used for high-quality, high-performance graphics rendering. The Pentium PCs equipped with optional graphics cards are used for low-end graphics rendering. In addition, as shown in the figure, this core VR computational and graphics facility is networked to the rest of the computer infrastructure at RTI and to extramural computer facilities such as the supercomputer from the Microelectronics Center of North Carolina.

Software:

The programming environment in the laboratory is Unix based. The SGI graphics workstations operate under the Irix operating system. Software packages used at RTI include Performer, Inventor, and Explorer. In particular, Performer provides a high-level application programmers' interface (API) for rendering the high-quality images which are characteristic of high-level VR applications. The IBM RS-6000 computers operate under AIX. The Apple Quadra PCs operate under the Apple OS operating system and the IBM PCs operate under the DOS operating system.

The programming model for the development of VR applications is illustrated in Figure 2. Using a variety of modelers and format translators, RTI has developed the capability of providing cost-effective VR solutions and to deploy them on the most appropriate platform from PCs to Silicon Graphics. One such approach is based on a rapid VR prototyping capability developed by RTI under an IR&D project and based on the low-cost modeler Virtus Walkthrough Pro®. Annotated polygonal databases are created in the modeler, either from electronic data or from drawings, and the results are transferred electronically into one of the Silicon Graphics workstations where the virtual environment is composed by the application of shading, textures, and light sources. The textures are derived from photographs of real objects scanned into the system through one of the IBM PCs. Textures can also be implemented procedurally. Once the environment is completed, interaction with the model is added using VR-DECK [6], a C-based software package developed by IBM T. J.Watson Research Center, in one of the RS-6000s. This package supports the instantiation and interaction of modules dedicated to specific functions in support of interactivity with and immersivity in the virtual environment. Thus, for example, there are modules dedicated to the capture of tracking and navigation data. These modules produce events which are used by a graphics-generating module to control the position and view of the camera model "searching" the data base. Additional features, such as speech recognition, sound generation, object behavior, etc., are available or can be added through additional modules. These modules can be distributed within one or many different workstations in the network, including the Reality Engines. This provides the ability to match the system resources to the application requirements.

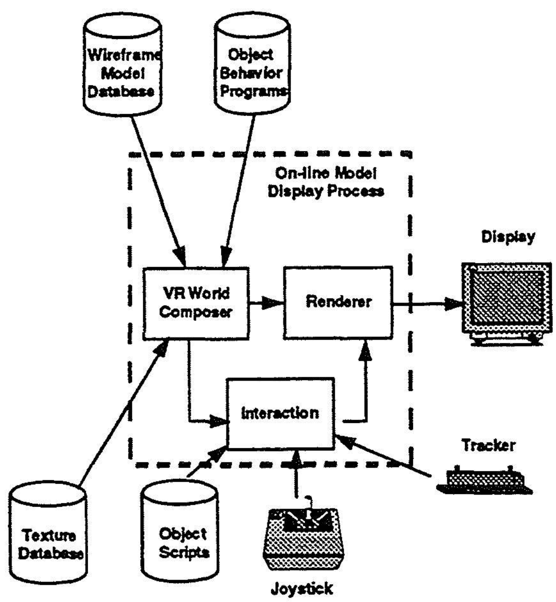


Figure 2 Conceptual Approach to the Implementation of the Virtual Environment

Another application development path in the laboratory uses AutoDesk's AutoCAD to capture or generate data for the models, 3D Studio to generate the models, and Sense8's WorldToolKit for development of the interactions with and immersivity in the resulting virtual environment. This environment is targeted toward the development of environments to be deployed on PC platforms. Current efforts have as a goal the creation of a seamless development environment in which models created by either modeler or from other non-RTI modelers (such as Intergraph's or ProEng) can be used in either the low- or high-end environment easily and cost-effectively.

In addition, the modular software system supports head tracking and stereo viewing to provide a trackable, immersive capability in the application. Further interactions with the virtual environments is achieved through the use of a speech recognition system operating in the RS-6000. Plans call for the development of software to support the integration of a 3D sound system and natural language processing into the applications as necessary.

VR PROJECTS

The Virtual Furniture Showroom:

For the Furniture Manufacturing and Management Center at North Carolina State University, RTI created and demonstrated the Virtual Furniture Showroom. This was a technology demonstration project for the furniture industry. Using drawings and sketches provided by American Drew Inc. and photographs of the real pieces of furniture, RTI developed virtual models of American Drew's Hancock Cherry bedroom collection, replicated synthetically the room in which the real collection was, and arranged the virtual collection and accessories in the room.

Interactivity with the furniture took on several forms. The visitor to the virtual showroom, donning a tracked HMD, could navigate over to a specific piece of furniture by gazing towards the piece and using a four -function joystick to walk over to or away from the piece. Then, by clicking on one of the other two buttons of the joystick, he/she could get a description of the piece through the earphones of the HMD. The application also supported the picking and moving of pieces of furniture as well as the changing of the finish on selected pieces within the collection.

Figure 3 illustrates the resulting virtual environment. This application was developed using Virtus Walkthrough Pro™ . Interaction and immersivity was obtained through VR-DECK. Rendering was done in an SGI Crimson and presented in a Virtual Research Eyegen3 HMD. Average scene complexity was on the order of 4,000 polygons with an average pixel depth of 5. The demands for high quality models required a large number of textures which, in some instances, overloaded the texture memory available in the Crimson, thus affecting rendering performance. Stereoscopic update rates on the order of ten per second were achieved when texture paging was not a factor. Update rates dropped substantially from this number when texture paging was a factor.The virtual showroom was demonstrated at the High Point, North Carolina, Fall International Home Furniture Market in October of 1993. The virtual showroom exhibit was located near the room with the real collection. This provided a way for the visitors to do a comparison while their impressions of the virtual room were fresh in their minds. Based on the reaction of the majority of the visitors, the virtual exhibit was a success. From a technology standpoint, it was concluded that VR technology was viable for the furniture industry in at least two areas: rapid prototyping of pre-market collections and interactive electronics catalogue of collections at high-end furniture stores. Use of this technology in furniture design and in many other applications should be feasible within the next five years as the cost of the technology continues to decrease and its performance continues to increase.

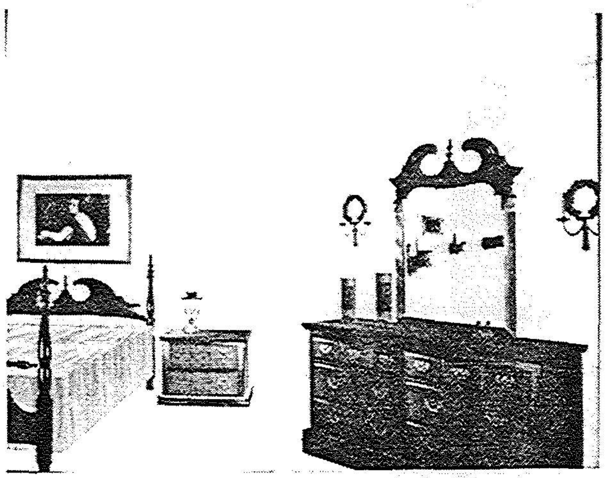


Figure 3 The Virtual Furniture Showroom

A Walkthrough of the Dresden Frauenkirche:

For two-hundred years, the Frauenkirche (Church of Our Lady) stood majestically over Dresden, Germany, as a magnificent example of Baroque architecture and an important expression of the Lutheran faith. On February 15, 1945, the church collapsed as a result of the intense heat produced by fire storms which resulted from extensive Allied bombings of the city. Today, efforts led by the Foundation for the Reconstruction of the Dresden Frauenkirche are under way to rebuild the church.

For IBM Germany, RTI developed an interactive and immersive walkthrough of the Dresden Frauenkirche. The goal of the project is to use the VR walkthrough as an exhibit where people can "visit" the church while it is under reconstruction. Figure 4 illustrates the interior of the Frauenkirche as it was in February, 1945, or as it will be in February, 2006. Thus, with the aid of Virtual Reality the visitor can step forward fifty years into the past and visit the magnificent Frauenkirche.

The interactive model was based on an animation model developed using the TDI modeling package [7]. This model had been derived from an original architectural model of the church done in the CAD software package CATIA™. The system used to implement the interactive and immersive walkthrough of the Frauenkirche consists of two workstations, an IBM RS-6000/650 and a Silicon Graphics Onyx Reality Engine, networked to accomplish the task. The tasks included tracking the orientation and navigation of the user, "the visitor" to the church, generating events resulting from the interpretation of the position data, updating the camera view of the database, generating a stereo pair of such view, and driving the two eye views of the head mounted display (HMD).

RTI scientists converted the animation model of the Frauenkirche (in TDI format) into a format compatible with the Performer™. Once this was done, application modules were implemented in VR-DECK and instantiated in the RS-6000. These modules included support of the head tracking operation, navigation, by means of the special joystick, and graphics. The latter invoked the rendering software developed for this application in the SGI Onyx. These modules were interconnected and activated in the RS-6000 according to the VR-DECK application protocol [8].

In particular, the graphics rendering module has the task of creating a view of the pictorial database (the church) as dictated by the head orientation and the user's position sensed by the tracker and indicated by the joystick, respectively. In the case of an immersive environment, this module also has the task of generating a stereo pair to support the presentation of a stereoscopic display in the HMD.

The interactive model of the Frauenkirche is one of the most complex models used to date to generate a virtual reality walkthrough application. (See Figure 4). The model consisted of 165,000 polygons, it made use of 12 textures in over twenty locations of the church, and was lit by five light sources. Scene complexity varies from approximately 80,000 polygons in the altar area to 20,000 polygons toward the back of the church with an average pixel depth of 4. Update rates varied depending on the direction of view with views of the altar updating at 3-5 frames per second (in stereo) and views of the balconies and the back of the church updating at 10-20 frames per second (in stereo).

Virtual Reality Models of the North Carolina Global TransPark:

For the North Carolina Air Cargo Airport Authority, RTI developed a series of immersive and interactive virtual models of the North Carolina Global TransPark. The NC Global TransPark is a bold initiative of the state of North Carolina to develop a multi-modal transportation facility built around the existing jetport in Kinston, NC. A development plan for the TransPark has just been completed. This plan calls for the development of the facility in stages and the models which RTI has implemented represent the various stages of development. Thus, the immersive models show the evolution of and allow planers to visit the concept from the present day to the project's conclusion in the next century. Figure 5 illustrates the resulting virtual models.

The three virtual models of the TransPark were built using the Virtus-based rapid prototyping capability. Unlike the case of the furniture showroom, however, existing CAD data was used extensively in the creation of the models. The first model shows the Kinston, NC, and its significant surroundings. Model 2 shows the intermediate development of the TransPark which includes a 13,000 feet cargo runway, a central cargo facility, a cargo transportation system, and assigned areas for the location of various activities such as office, research, and light industrial activities. Model 3 includes the complete vision of the TransPark which contains a second runway parallel to the original one. A control panel associated with each model allows the visitor to look at the existing wetlands and to visualize how they will change as the development takes place. In addition, time of day and visibility conditions can also be controlled through the panel. Included in all models are "dumb" agents representing activities which will take place in the Park. These include airplanes taking off and landing, trucks moving along highways, and a train moving along the railroad tracks.

There are several display options associated with this application. The models can be shown in a stereoscopic, augmented reality mode either on a monitor screen using shuttered glasses or on a projection screen using passively polarized glasses with no head tracking and mouse-based navigation; or they can be shown in a HMD with head tracking and joystick-based, direction-of-gaze navigation.

The models range in scene complexity from 4,000 to 10,000 polygons with an average pixel depth of 3. Stereoscopic update rate performance varies from 10 to 20 updates per second. Texture paging was not a factor in this application and we also were able to use level-of-detail models to optimize performance.

Maintenance Training Simulator-National Guard (MTS-NG):

For the Advanced Research Projects Agency, RTI has designed and implemented an advanced training system for home-station training of National Guard tank mechanics. The maintenance training simulator for the National Guard (MTS-NG) is a computer-based instructional system which uses Virtual Reality as the human-computer interface between the trainee (mechanic) and the training system, significantly extending training to personnel at sites without equipment. The MTS-NG integrates VR, multimedia and instructional technologies to provide training to tank turret mechanics (45T) to perform diagnostic and maintenance on M1A1 Abrams Tank and M2A2 Bradley Fighting Vehicle. Figure 6 shows the MTS-NG development team testing the various stages of the advanced instructional system.

This application has been implemented in a 90 MHz Pentium PC equipped with a SPEA Graphiti Series Fire graphics board and a StereoGraphics Corporation's CrystalEyes PC for stereoscopic image generation and viewing. The software development environment consists of Autodesk's 3D Studio modeler for the generation of the databases, Sense8's WorldToolKit for the building of and interaction with the virtual environments, and MicroMedia's Authorware for the generation of the courseware.

The courseware includes the lessons used in the Regional Training Site (RTS). The courseware launches the virtual reality applications when appropriate to the purpose of the lesson. These virtual environments include navigation through and interactions with solid models of the two vehicles as well as cutaways of their interior showing all the Line Replaceable Units (LRUs). They also include the ability to interact with the interior of the gunner's compartment and of the driver's compartment. Using this interactivity, the student may select any of the LRUs in either compartment for closer inspection and in turn switch switches, rotate knobs, etc., which can be used in performing diagnostic tests when used in conjunction with the Simplified Test Equipment (STE) and under the supervision of the training module. The training module guides and/or monitors the students' progress in diagnosing a fault in the vehicle. The student uses military training manuals to perform a series of interactive tests in a manner identical to the ones in the real vehicle.

Scene complexity varies from about 2,000 polygons for the external 3D views of the vehicles to 15,000 polygons for the interior view of the gunner's compartment with an average pixel depth of 3. These are textured polygons. Stereoscopic update rates vary from 1-2 frames per second for the most complex scenes to 10-15 frames per second for the less complex scenes. Texture memory available in the FIRE graphics board (8 MBytes) accommodates the textures used in this application. Anticipated hardware and software performance improvements will improve current performance. Also, if necessary, improved performance can be attained with a higher-performance platform. This will require a tradeoff between cost and performance since one of the design goals was to deploy this system for less than $10,000 per copy. Total system cost is $9,600 per unit.

CONCLUSIONS

Our work and that of many other applied researchers have demonstrated that VR is a viable technology with serious applications in areas other than low-end entertainment (arcades) and high-end simulators. When should one use VR? When the value of applying it exceeds the cost of developing it and also when it supports, enhances, and improves current or anticipated practices. It is our experience that development of a detailed requirements definition as a first step of a VR project leads to the development of cost effective solutions of VR problems. The bottom line is that virtual environments should not be more real than necessary for the application. Hardware and software performance improvements will continue to support more and more sophisticated applications of VR at lower and lower cost. As it was with PCs in the early 1980s, this will lead to the democratization of VR and the proliferation of its applications. We also anticipate wide use of VR technologies in the rapid product prototyping arena.

ACKNOWLEDGEMENTS

The VR group at RTI wishes to acknowledge the contributions of Mr. James B. Clary, Vice-President, Electronics and Systems, at Research Triangle Institute for his vision and unwavering support of the effort to develop a VR capability, Dr. Robert F. Helms, II, for his indefatigable efforts to make VR a viable business at RTI, and Mr. Dale W. Rowe for devoting his superior technical management skills to the stewardship of the effort. To the members of RTI's Center for Digital Systems Engineering which labored countless hours to implement the VR laboratory and make the projects described in this paper a success, thanks! They are: Jim Aanstoos, Jim Chung, Stephen Duncan, Susan Grantlin, Dean Hering, Dave Mclin, Jim Morrison, Daniel Nissman, Maria Ward, and Hal Waters. Our thanks go also to the members of the Virtual Worlds Group at IBM's T. J. Watson's Research Center and in particular to its former manager, Bryan Lewis.

REFERENCES

1. Britt, C. Les and Montoya, R. Jorge, et al, "Cockpit Display of Traffic Information (CDTI) Experiment Design Task," NASA CR-145340, National Aeronautics and Space Administration's Langley Research Center, Hampton, VA, May, 1978.

2. Montoya, R. Jorge, England, J. Nick, and Rajala, Sarah A., "An Advanced Programmable/ Reconfigurable Color Graphics Display System for Crew Station Technology Research," Proceedings of the 4th AIAA/IEEE Digital Avionics Systems Conference, St. Louis, Missouri, November 17-19, 1981, pp. 486-498.

3. Montoya, R. Jorge, Lane, Harold H., Turner, Tim L., and Hatfield, Jack, J., "The Application of a Color, Raster Scan, Programmable Display Generator in the Generation of Multiple Cockpit Display Formats." Presented at the AIAA/IEEE 5th Digital Avionics Systems Conference, Seattle, Washington, October 31-November 3, 1983.

4. Montoya, R. Jorge, Turner, Tim L., Suresh, Ramasubramanian, and Williams, Darryl A., "An Interactive Graphics Editor for Computer-Generated Cockpit Displays," Proceedings of the 9th AIAA/IEEE Digital Avionics Systems Conference, Virginia Beach, Virginia, October 15-18, 1990, pp. 441-446.

5. Montoya, R. Jorge, Turner, Tim L., Jewell, Donna M., Aanstoos, James V., Barker, M. Chad, and Williams, Darryl A., "An Ada-based, Portable Design Workstation for Computer-Generated Cockpit Displays," Proceedings of the 9th AIAA/IEEE Digital Avionics Systems Conference, Virginia Beach, Virginia, October 15-18, 1990, pp. 582-588.

6. Codella, Chris, Jalili, Reza, and others, "A Toolkit for Developing Multi-User, Distributed Virtual Environments," Proceedings of VRAIS 1993, September 1993, pp. 401-407.

7. Collins, Brian, "From Ruins to Reality - The Dresden Frauenkirche," IEEE Computer Graphics and Applications, November 1993.

8. Jalili, Reza, Kirchner, Peter, Montoya, Jorge, Duncan, Stephen, and others, "A Visit to the Dresden Frauenkirche," Research Report, RC 19662 (87190), IBM T. J. Watson Research Center, July, 21, 1994.

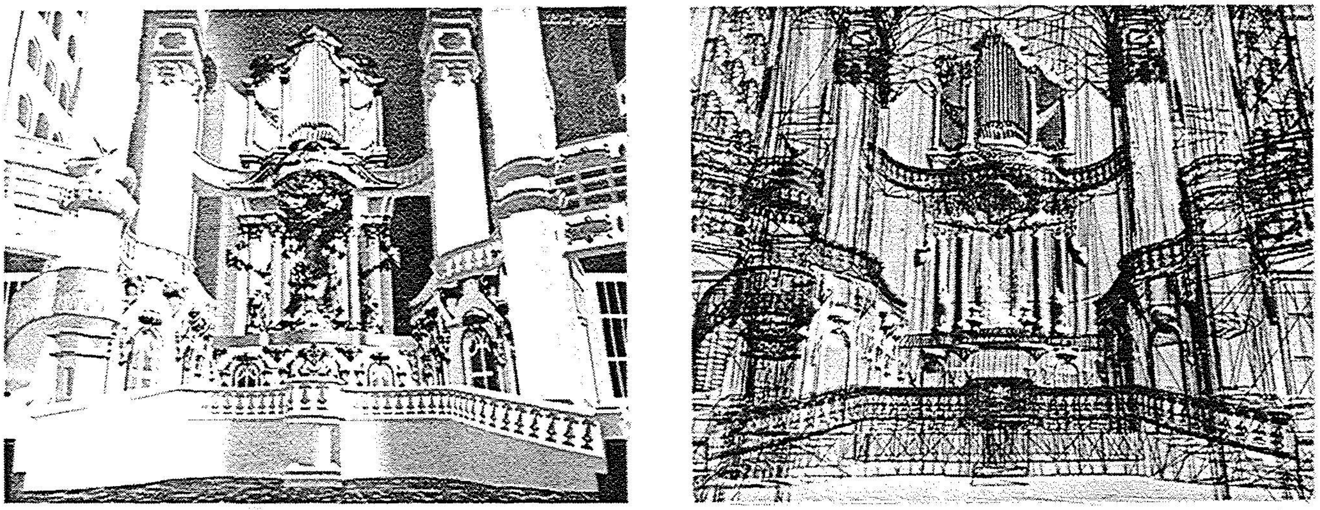


Figure 4 VR Model of the Dresden Frauenkirche

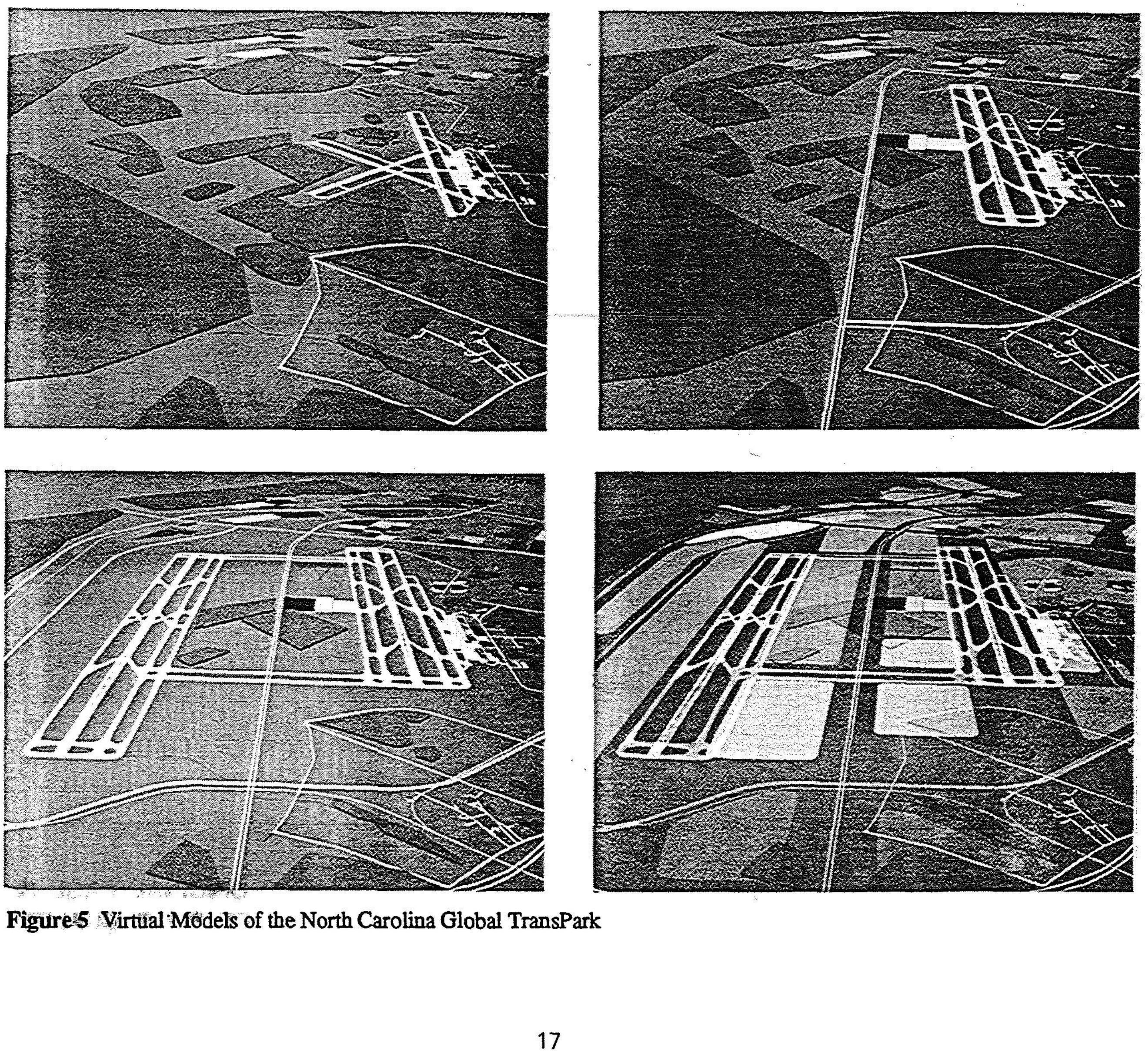


Figure 5 Virtual Models of the North Carolina Global TransPark

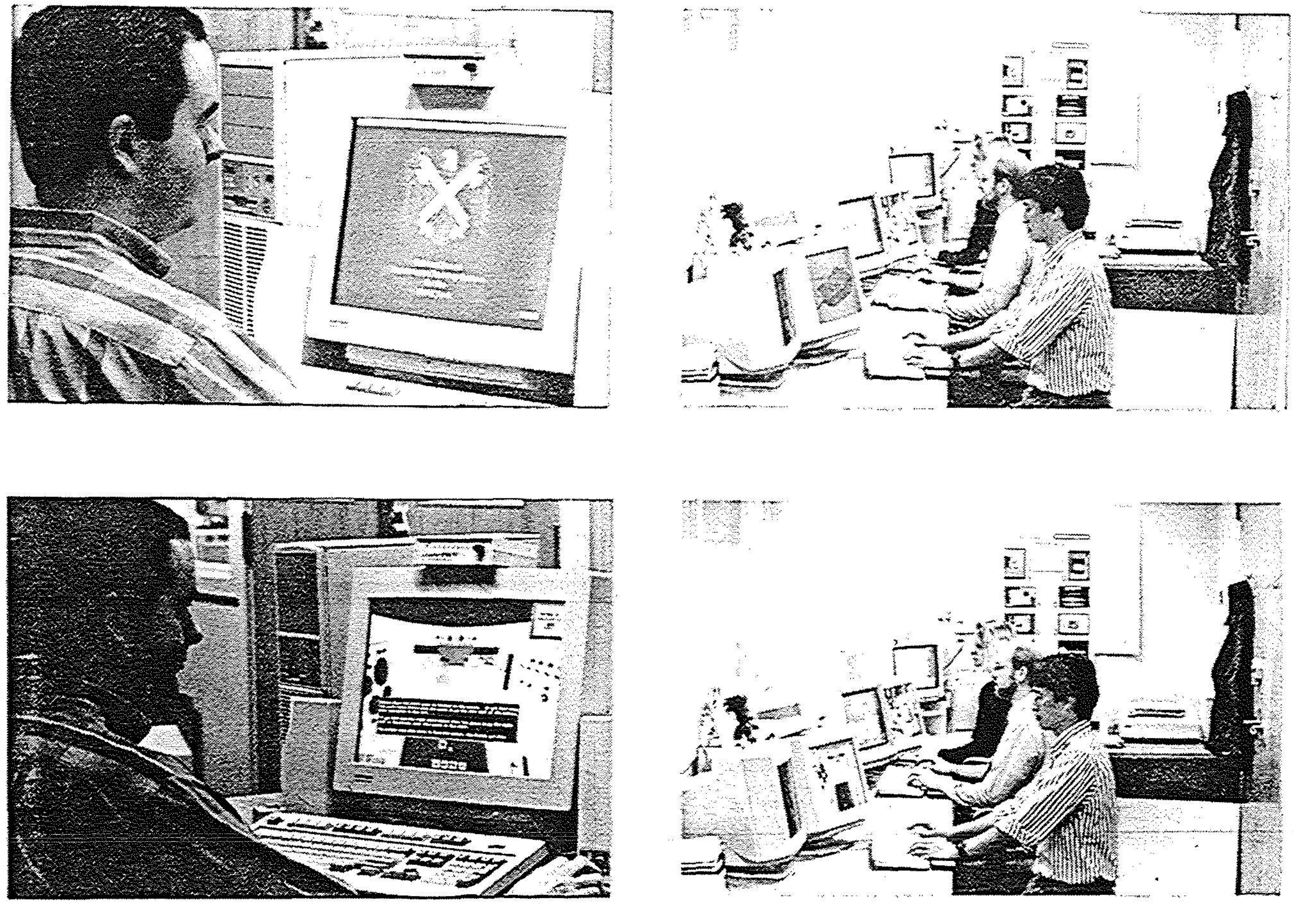


Figure 6 Advanced Instructional System for the Maintenance Training Simulator for the National Guard

Session 2
Applications

Simulation of Arthroscopic Surgery Using MRI Data

Geoffrey Heller and Jon Genetti

Arctic Region Supercomputing Center
University of Alaska
Fairbanks, Alaska, 99775-6020
e-mail address: heller@arsc.edu, genetti@arsc.edu

ABSTRACT

With the availability of Magnetic Resonance Imaging (MRI) technology in the medical field and the development of powerful graphics engines in the computer world the possibility now exists for the simulation of surgery using data obtained from an actual patient. This paper describes a surgical simulation system which will allow a physician or medical student to practice surgery on a patient without ever entering an operating room. This could substantially lower the cost of medical training by providing an alternative to the use of cadavers.

The project involves the use of volume data acquired by MRI which is converted to polygonal form using a corrected marching cubes algorithm. The data is then colored and a simulation of surface response based on springy structures [8] is performed in real time. Control for the system is obtained through the use of an attached analog-to-digital unit. A remote electronic device is described which simulates an imaginary tool having features in common with both arthroscope and laparoscope.

INTRODUCTION

After consultation with persons in the medical profession we have decided to build a system to simulate arthroscopic surgery on the human knee. Of particular interest are the sports-related injuries which are becoming more and more common. Some reasons for this decision are:

- The surgery is relatively simple.
- Video of this type of surgery is readily available.
- MRIs are already regularly used to diagnose patients who may later need arthroscopic surgery.
- MRI data sets of human knees are relatively common (although they can be difficult to obtain due to legal barriers.)
- The surgery is usually done using a remote viewing device (an arthroscope) attached to a monitor.
- The surgery is generally not life-threatening.

We are currently in the process of developing the surgical simulation system described here. Our software presently runs on a Silicon Graphics Onyx. An attached electronic device provides three dimensional input to a real-time display program running under X and OpenGL. The simulation can also be run on an Indigo 2, provided that the workstation has sufficient memory.

It is our intention that the initial version of the software be targeted primarily as an educational tool for use in medical schools until any bugs which exist can be worked out. By allowing students to practice surgery on a simulation they can be better prepared for their first surgery on an actual cadaver. This will also allow for the students to spend more time practicing surgery before entering the professional world.

BACKGROUND

The simulation of surgery brings into focus problems from a broad set of disciplines. One of the difficulties which exists is in obtaining data appropriate for use in the simulation. Fortunately for us, physicians already commonly employ a scanning device called a magnetic resonance imager in diagnosing patients for arthroscopic surgery.

An MRI is a scanning device which uses electromagnetic radiation to create a series of two-dimensional images of the human body. By placing stacks of MRI images together we obtain a volume data set which can be used in rendering a three-dimensional image. While the images produced by MRI show tissues in a manner which is easily recognizable to the human eye, the data cannot be easily interpreted by computer. Unlike Computerized Axial Tomography (CAT) scans, the data from an MRI does not represent tissue density but rather the quantity of residual electromagnetic radiation present after the initial source is removed.

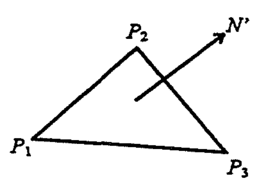
As the MRI data does not contain tissue type information (in any direct sense) it is not possible for an algorithm to make a binary decision to discern two types of tissue from one another. The solution, although unattractive, is to have a professional radiologist "color" all of the plates manually. By coloring, we mean that the individual must identify all tissues on each plate and assign a value. Once this process is completed we can proceed to a marching cubes rendering of the image.

Why use the marching cubes algorithm? In order to create a convincing simulation we need to be able to achieve a certain frame rate in the animation. So far, the most powerful animation hardware today requires the use of polygonal objects. Also, the marching cubes algorithm creates a very regular distribution of vertices which we will see later on is useful in our animation process.

Facet Rendering of Marching Cubes Output

One of the problems with the marching cubes algorithm is that it does not reliably produce correct tracing directions for the vertices in output triangles [10]. The resultant erroneous normals produce distracting "holes" if the image is rendered using a facet algorithm. We present here a simple solution to this problem that fixes triangles which were traced incorrectly by comparing their surface normals to the gradients present in the original volume data.

We start with the general formula for computing the surface normal of a triangle. Notice that the equation is sensitive to the direction in which the vertices are traced. Here the trace direction determines if the normal is heading into or out of the screen:



$$N' = \frac{(P_3 - P_1) \times (P_2 - P_1)}{\|P_3 - P_1\| \cdot \|P_2 - P_1\|}$$

Fig. 1: Surface normal using three vertices.

Now we look at the set of equations describing the surface normal computation using a 6-point gradient:

$$N.x = \frac{G(i+1,j,k) - G(i-1,j,k)}{2}$$

$$N.y = \frac{G(i,j+1,k) - G(i,j-1,k)}{2}$$

$$N.z = \frac{G(i,j,k+1) - G(i,j,k-1)}{2}$$

Fig. 2: Surface normal using a 6-point gradient.

We now use the gradients to verify the trace direction for marching cubes output triangles. For each marching cubes triangle T, calculate the surface normal N' as in Figure 1 and N as in Figure 2. After normalizing both normals we take the dot product of the two and do the following:

if $S \geq 0$ then do nothing.
if $S < 0$ then reverse the trace direction of vertices in triangular face T.

The above technique is an after-the-fact method for repairing triangles which were traced incorrectly by the marching cubes algorithm. By making sure that all facets are traced in the right direction we will be able to use this information to calculate proper surface normals quickly during the rendering phase of animation.

Arthroscopic Surgery

One of the principal problems in simulating surgery is that the vast medical field comes into play. There are as many surgical techniques as there are types of injury. In order to limit our scope to a practical level, we must think in terms of modeling the human body as opposed to creating a system for teaching correct surgical technique. Here the specifics of surgical tools and technique are not the issue so much as being able to create a realistic simulation. Only after a realistic simulation of the human body has been developed can we consider the specific techniques of surgery.

Earlier we mentioned that we are developing an electronic data acquisition system that will simulate features of both laparoscope and arthroscope. Note that arthroscopes are used in surgery on the joints while laparoscopes are used in abdominal surgery. While the arthroscope is simply a fiber optic viewing device, some laparoscopes feature surgical implements which are remotely controlled by the surgeon. It is these controls which I intend to simulate with the electronic device. The computer's monitor will serve to simulate the arthroscope's viewing screen. The reason for this is that the remote control limits the degrees of freedom for the user and therefore makes development of the user interface less complex.

Why not simply simulate laparoscopic surgery? Unfortunately the tissues involved do not show up well on either CAT or MRI scans due to the tendency of the patient to move (breathe) while being scanned.

THE PROCESS OF SURGERY SIMULATION

The process of surgery simulation has a great deal to do with animation. Here we present a procedure for doing the simulation which should be able to run at a good frame rate:

1. System determines the *primary vertex* from the user selection by using a *three-dimensional hashing function.*
2. The *region selection function* uses the tissue type of the primary vertex to determine which neighboring vertices are in the *primary active region.*
3. The *push-through determiner* decides if there is sufficient force being applied to an object to warrant the creation of a *secondary vertex.*
4. If a secondary vertex is created, the region selection function is again called with a parameter to create a *secondary active region* which is generally smaller than the first. (No tertiary vertices are considered.)
5. The active regions are handed over to the *springy surface animation algorithm* which uses tissue type to adjust the spring properties.
6. The user's tool selection determines the control operator for the springy surface animation algorithm. (touch, grab, nibble)
7. Surface is modified for this frame, normals are recomputed and the frame is sent to the screen.

Selection of the Primary Vertex

Every movement that the user makes in three dimensional space must have an associated collision test to determine if a region should become active. The problem of doing these collision tests in real time is that we often have an large set of vertices with which to compare. Fortunately there is a simple solution involving the use of a three dimensional hashing function:

$$hash(i,j,k) = i + j \cdot x\max + k \cdot x\max \cdot y\max$$

For every point *(i, j, k)* selected by the user we use the three dimensional hashing function *hash* to generate a pointer to the hashing table which in turn contains pointers to vertex table entries.

The hashing table was generated during the initialization phase of the program by going through the vertex table and using each vertex point as a parameter to the hashing function. The hashing table is then given an entry containing a pointer to the original vertex table entry.

The Region Selection Function

Region Selection Functions (RSFs) are routines which determine the region to become active using a starting vertex and its coloring information. All RSFs work by spreading from the starting vertex in all directions on the surface. By tracing edges to new vertices we are able to prevent jumping to tissues which are near the active region but should not be affected.

The RSFs differ in how they determine when to stop spreading out. For example, certain types of long muscle would have an RSF that is an oval shape oriented along the length of the muscle. Other RSFs might describe regions whose influence is purely circular. When the vertices being traced have spread to the boundaries they are marked as being "nailed" for later use in the springy surface animation algorithm. A nailed vertex will not move during the springy surface animation. This is a necessary simplification as the springy surface computations need to occur relatively quickly.

The Push Through Determiner

After a user has selected a primary vertex and has moved it from its original position the possibility exists for a *secondary vertex* to come into play. The secondary vertex in necessary to describe the effect on the other side of a soft object when the force applied on the original vertex has passed through the object.

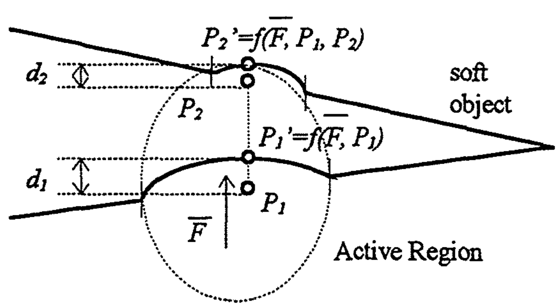


Fig. 3: Determining secondary active regions.

To determine a secondary vertex we draw a line from the original position of primary vertex P_1 to the current position of primary vertex P_1' as seen in Figure 3. The length of this line corresponds to the amount of force that the user has applied. We then extend the line by using the magnitude and sign of the applied force to determine the respective amount and direction of extension. By using preselected active regions we do effectively limit the amount of force we can simulate being applied to a vertex. This should not be a problem as it is unlikely that a surgeon will be needing to exert excessive force in the course of using the simulation.

Note that there is no absolute guarantee that the first similar surface encountered is part of a single object. However, in the case of human knee data sets it is highly probable that this secondary surface is part of the same object. The limits imposed by the region selection functions serve to further prevent uninvolved surfaces from being animated.

Secondary Active Regions

If the push through determiner has located a secondary vertex it then becomes necessary to add a secondary active region. This secondary active region is computed using rules similar to that of the primary active region but with one exception. The area of the secondary active region is a function of the amount of force being applied and the distance between the primary and secondary vertices.

The Springy Surface Animation Algorithm

The main engine of the simulation is a springy surface algorithm which follows along the work of Haumann [8] at Ohio State University. In our tissue simulation model we have two distinct sets of springs which we are animating. The first set of springs exists along each edge in the selected region. The individual springs are axially springy but radially rigid. Each edge shared by two triangles forms a hinge. Unlike many springy surface models, there is no spring holding hinged triangles in place. We instead have springs between the current vertex positions and their original positions. Under this model the objects being simulated will have an affinity for their original shape. Unless a tool is being used which calls for a permanent shape change, all vertices will eventually return to their original configuration.

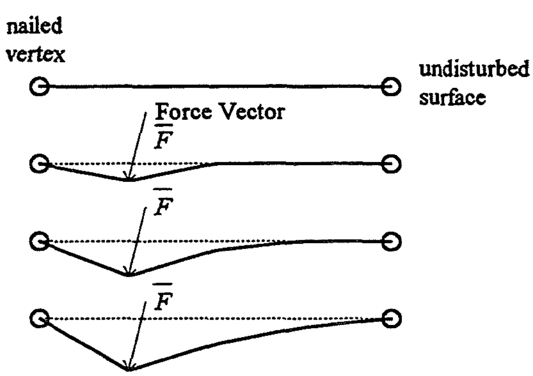


Fig. 4: Applying force to a springy surface.

Another difference between our springy surface model and the more commonly used ones is that our vertices are massless. The scale on which the surgery is taking place is so small that tissues respond as if they had no mass at all. We are able to take advantage of this by using massless vertices to lighten the computational overhead.

Tools of Surgery

For the purpose of our simulation there are essentially only three tools which can be used. There is the probing tool which allows the user to poke and press into an area, the clamp or grabber tool which allows the user to both push and pull at an area, and the nibbling tool which "removes" tissue at the specific area.

While the probe tool allows the user to push along a surface causing dynamic reallocation of the selected regions, the grabber tool forces the user to stay in the selected region until the grip is released. Furthermore, the nibbling tool actually does not cut but merely pushes vertices away from the tool. This simplification prevents us from having to dynamically reconstruct the connections between triangular faces.

Display Phase

After the new vertex positions have been computed all that remains is to determine the new surface normals for the active regions and send the data to the renderer.

THE ARTHROSCOPE SIMULATION DEVICE

We describe a simple microcontroller circuit which can be used interface analog controls to a host computer. Based on Motorola's 68HC11 microcontroller, the circuit described here is an eight-bit, eight-channel analog to digital converter featuring an optional status display. Data is sampled from up to eight independent analog sources and is sent via serial connection to the host machine.

The device is intended to be operated in a polled mode where the host system transmits a sample request for a particular analog device numbered 1 through 8. The microcontroller then decodes this command, performs the sample, and sends the resultant information back to the host.

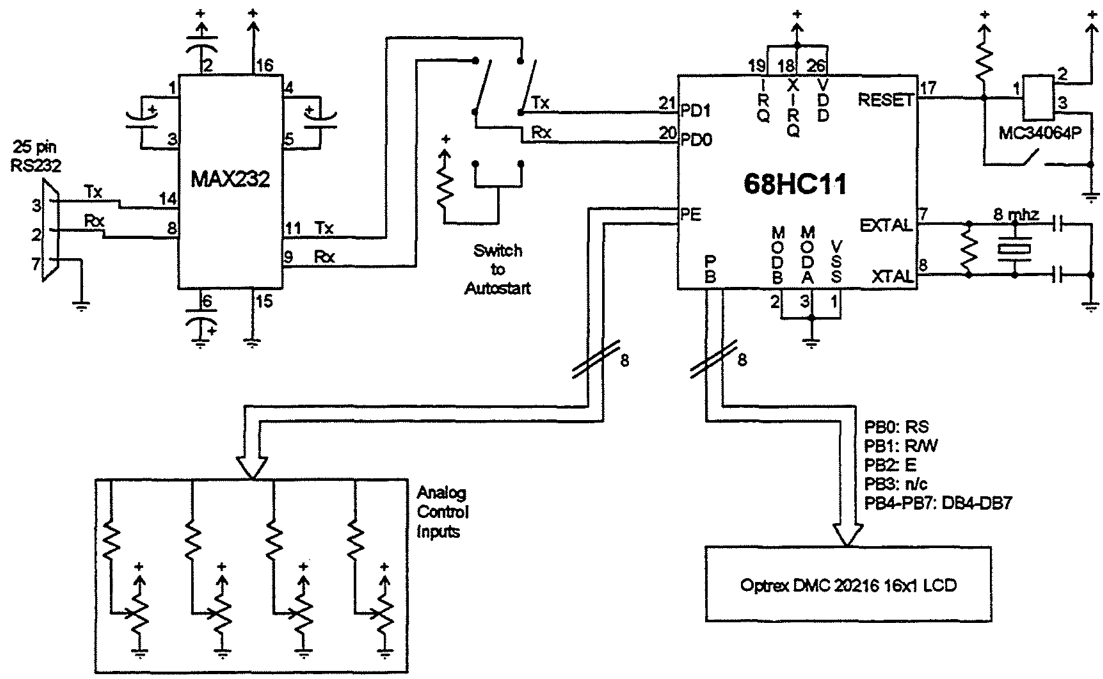


Fig. 5: Simulation control device.

FUTURE WORK

One of the main drawbacks of the method described above is that it limits the area which can have the spring algorithm applied to it. This limitation is necessarily imposed due to the processing capacity of single-CPU computer systems. However, there is the possibility of a much more accurate simulation if all vertices could be included in the springy surface algorithm simultaneously. With this in mind we are working on an interface for the Cray T3D massively parallel processor system. In this model the T3D would run springy surface calculations while a Silicon Graphics Onyx would render the output. A high-speed HIPPI connection between the two machines should provide enough bandwidth to perform the simulation.

Expanded computing power might also allow for the implementation of multiple region selection. With this we would be able to simulate a tool interacting with a surface at more than one point. This would increase the amount of realism in our simulation.

Another area of improvement under consideration is the use of a Gouraud shading model. Since the only additional requirement of this model is that we compute normals for each vertex it should be possible to add this feature without major modification to our source code.

CONCLUSION

By using scanned data we have greatly increased the level of realism in surgery simulation over systems which use mathematical tissue models. In order to limit the explosive computational complexity we have at many points opted for efficient simulation algorithms over algorithms which produce realistic simulations. What we gain by the trade off is a decent frame rate for our animation. A quality simulation means nothing if the frame rate is so low that the system becomes unusable.

A wide range of expertise is needed to successfully develop a system for the simulation of surgery. While technical problems often have obvious solutions, these solutions do not necessarily provide for the best possible simulation. In order for a quality system to be developed it is necessary to do extensive consultation with medical professionals and a review of the equipment and procedures involved.

ACKNOWLEDGMENTS

Randy Seidlitz and G.E. medical systems for providing MRI data sets of human knees.

Greg Schmidt and Jim Snell of Texas A&M University for their VETIT program which allows for the coloring of MRI plates.

This research was supported in part by Cray Research, Inc. and the Strategic Environmental Research and Development Program (SERDP) under the sponsorship of the Army Corps of Engineers Waterways Experiment Station.

REFERENCES

[1] Baraff, D. and Witkin, A. "Dynamic Simulation of Non-penetrating Flexible Bodies," Computer Graphics, vol. 26(2), July 1992, pp. 303-308.

[2] Chadwick, J., Haumann, D., and Parent, R. "Layered Construction for Deformable Animated Characters," Computer Graphics, vol. 23(3), July 1989, pp. 243-252.

[3] Chen, D. and Zeltzer, D. "Pump It Up: Computer Animation of a Biomechanically Based Model of Muscle Using the Finite Element Method," Computer Graphics, vol. 26(2), July 1992, pp. 89-98.

[4] Clemente, C. Gray's Anatomy. 30th American Edition, Lea & Febiger, 1985, pp. 107-113., pp. 397-408.

[5] Delp, S., Loan, P., Hoy, M., Zajac, F., Fisher, S. and Rosen, J. "An Interactive Graphics-Based Model of the Lower Extremity to Study Orthopaedic Surgical Procedures," IEEE Transactions on Biomedical Engineering, vol. 37(8), 1990.

[6] Gourret, J., Thalmann, N., and Thalmann, D. "Simulation of Object and Human Skin Deformations in a Grasping Task," Computer Graphics, vol. 23(3), July 1989, pp. 21-30.

[7] Green, M. "The $15.00 Wonder Computer," Portland Area Robotics Society (PARTS), Issue #5, (February 1993), pp. 5-6.

[8] Haumann, D. "Physically Based Modeling of Springy Structures," Ohio State University, (Lecture Notes), 1987.

[9] Jones, J., Flynn, A. Mobile Robots: Inspiration to Implementation, A K Peters, 1993, pp. 93-138.

[10] Nielson, G., Hamann, B. "The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes," IEEE Computer Society Press, 1991, pp. 83-91.

[11] Pieper, S., Rosen, J., and Zeltzer, D. "Interactive Graphics for Plastic Surgery," Computer Graphics, Annual Conference Series, 1992, pp. 127-134.

[12] Pommert, A., Bomans, M., and Hohne, Karl. "Volume Visualization in Magnetic Resonance Angiography," IEEE Computer Graphics and Applications, vol. 12(5), September 1992, pp. 12-13.

[13] Sagar, M., Bullivant, D., Mallinson, G., and Hunter, P. "A Virtual Environment and Model of the Eye for Surgical Simulation," Computer Graphics, Annual Conference Series, 1994, pp. 205-212.

[14] Terzopoulos, D., Platt, J., Barr, A., and Fleisher, K. "Elastically Deformable Models," Computer Graphics, vol. 21(4), (July 1987), pp. 205-214.

[15] Waters, K. "A Muscle Model for Animating Three-Dimensional Facial Expression," Computer Graphics, vol. 21(4), (July 1987), pp. 17-24.

[16] Weil, J. "The Synthesis of Cloth Objects," Computer Graphics, vol. 20(4), August 1986, pp. 49-54.

Multimodal Correlation and Intraoperative Matching of Virtual Models in Neurosurgery

Enrico CERESOLE, Michele DAL SASSO and Aldo ROSSI
DIMEG - Department of Innovation in Mechanics and Management
University of Padova
Via Venezia 1 - 35131 Padova - ITALY
Phone: ++39(49)8286820 Fax:++39(49)8286816
e-mail: gaspare@hpdimeg.unipd.it

ABSTRACT

The multimodal correlation between different diagnostic exams, the intraoperative calibration of pointing tools and the correlation of patient's virtual models with the patient himself, are some examples, taken from the biomedical field, of a unique problem: determine the relationship linking representation of the same object in different reference frames. Several methods have been developed in order to determine this relationship, among them, the surface matching method is the one that gives the patient minimum discomfort and the errors occurring are compatible with the required precision. The surface matching method has been successfully applied to the multimodal correlation of diagnostic exams such as CT, MR, PET and SPECT. Algorithms for automatic segmentation of diagnostic images have been developed to extract the reference surfaces from the diagnostic exams, whereas the surface of the patient skull has been monitored, in our approach, by means of a laser sensor mounted on the end effector of an industrial robot. An integrated system for virtual planning and real time execution of surgical operation, that is described in this paper, has been realized at the Department of Mechanical Innovation and Management of the University of Padova, Italy, in cooperation with the Neurosurgical Division of the Hospital of Treviso, Italy.

1. INTRODUCTION

In recent years, systems for surgical planning and intraoperative assistance have been widely used. In these systems a CT or MR exam is used to reconstruct a virtual model of the anatomies of interest. Dedicated software programs allow the surgeon to simulate on the virtual model the approach strategies, to determine and study the target points, the trajectories, etc. Several of these systems allow the correlation among different diagnostic exams. To correlate different diagnostic exams it is necessary to compute the transformation matrix between the reference frames associated to each diagnostic modality. A classical example of it in the neurosurgical field is the stereotactic head frame. Several types of localizers, mounted on the base ring of the head frame, fixed to the patient's skull, produce artifacts in the diagnostic images. Localization of the artifacts in the images allow to calculate the coordinates of each voxel of the scanned volume with respect to the head frame. In this way a correlation can be found, between a diagnostic image and the stereotactic system. Using different types of localizers, this correlation can be established between each diagnostic modality and the stereotactic reference frame, and consequently among the diverse diagnostic exams. In more recent years, newer correlation methods based on artifacts produced by fiducial markers fixed to the patient's skin or implanted in the bone were developed. The data gotten from the planning stage can be used during surgery only if a registration method is found, correlating the reference frame of the "real" patient with one of the diagnostic exams. Such a registration is implicitly achieved if a stereotactic head frame is used, while if fiducial markers are employed it is necessary to use a three-dimensional digitizer to determine the position of the fiducial markers in the real world. A correlation matrix can be computed by surface matching only if equivalent reference surfaces can be extracted from different diagnostic exams. Among all the surfaces, the CT is taken as a reference, because it is characterized by very low distortion. The same procedure can be applied to intraoperative matching, determining, by means of suitable sensors, the real surface to be correlated to the CT. In this paper methods for image acquisition and processing for multimodal correlation and intraoperative matching by means of surface matching techniques are described. These methods are general enough to be applied to the calibration and adjusting of robots in other fields than the neurosurgery.

2. MULTIMODAL IMAGING

The features of each diagnostic exam are due to different physical quantities that are sampled. For instance, the CT provides a great deal of information on the bone tissues, but its spectrum is very narrow for soft tissues (about 100 Hounsfield numbers out of 2001). On the other side, MR images show soft tissues very efficiently, but bones are not visible. Blood vessels are usually detected using Digital Angiographs. Even the most expert radiologist would find hard to mentally correlate the information gotten from all these most expert radiologist would find hard to mentally correlate the information gotten from all these diagnostic exams. For this reason our research was aimed at finding a non invasive method, which could guarantee high reconstruction precision, easy to be implemented and general enough to be applied to different anatomical districts and different diagnostic modalities.Our work started off using fiducial markers fixed to the patient's skin and visible in all diagnostic exams; in this case the correlation between the different modalities was calculated on the basis of the position of the markers in the various reference frames. Algorithms for 3D-3D and 2D-3D correlation were developed and tested. Then, the surface matching was introduced as a method to minimize the transformation errors. Such a hybrid approach (surface matching + fiducial markers) revealed itself adequate for diagnostic purposes, and can be effectively used also for intraoperative matching.

3. INTRAOPERATIVE MATCHING AND ASSISTED EXECUTION

The virtual model of the patient is used by the surgeon to define the targets, the approach directions and more generally the surgical tasks to execute thru the robotized system. The system is made of an ASEA IRB2000 industrial robot on which a laser system for distance measurement (by Microepsilon) has been mounted, a robotic simulation software (Robcad, by Tecnomatix), and a three-dimensional digitizer (Surgicom by Faro Tech.). The digitizer is a six d.o.f. passive arm connected to a UNIX workstation thru a serial interface box. The operating room is thoroughly modeled in the robotic simulation environment. The environment is totally structured except the volume occupied by the patient and by the mobile medical devices associated to him. To guarantee the patient's safety, once he has been fastened to the operating bed thru a Mayfield (Ohio Instr.) head holder, the position of the fiducial markers on the patient's skull is determined by means of the three-dimensional digitizer.

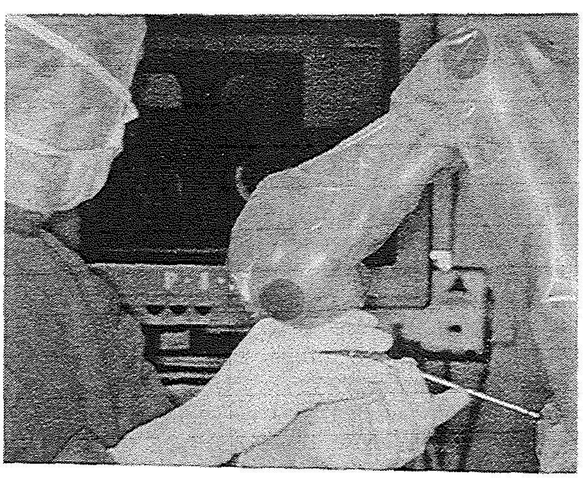


Figure 1- Registration of the fiducial markers.

The coordinates of the fiducial markers are transformed from the digitizer's reference frame into the robot's reference frame by means of a transformation matrix which was computed during the cell definition procedure. In fact, the cell has been defined a priori localizing all object inside it, including the digitizer, with respect to the robot's reference frame. This allows to define a bounding security volume that may never be entered by the robot and the instruments associated to it. Moreover, in order to avoid interference between the robot and the devices inside the operating cell, all movements the robot is to make for the registration procedure are simulated before execution. The data of the patient's reference surface are gotten by means of a laser sensor installed on the robot's end effector. This acquisition procedure is firstly simulated utilizing the virtual surface extracted from the diagnostic images as a reference, so that the laser beam is kept perpendicular to the surface itself. Then, during the measurement procedure in the real world, the direction of the laser beam is adjusted on the basis of the previous measurements. Therefore, the measurement procedure in the real world does not correspond exactly to the simulated one, but there are minor differences in the robot's end effector positioning.During the simulation stage, the direction of the first approach to the patient, to begin the measurement scanning, is defined. Knowing then the position of the fiducial markers, computed using the digitizer, an approximated transformation between the patient's and the robot's reference frame can be found. This function is used to limit the number of iterations of the surface matching algorithm, avoid the problem of local minima and overcome the ambiguities due to the symmetry of the skull.

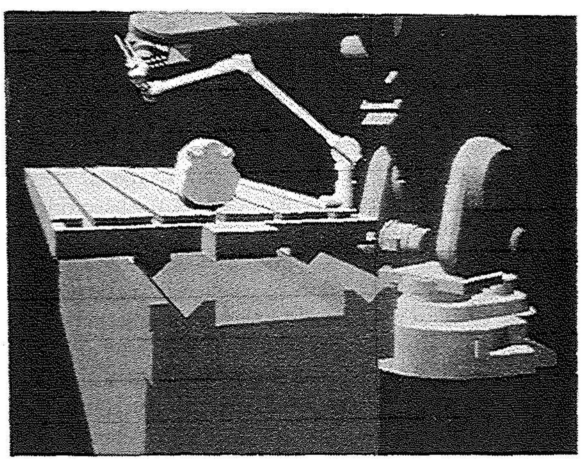


Figure 2 - The operating cell model with the 3D-digitizer, the robot and the CT surface during the registration procedure.

The surface matching algorithm computes the patient-robot transformation minimizing the distance between the reference surface extracted from the diagnostic images and the real surface measured by means of the laser sensor. This matrix is then used to transform into the robot's reference frame al the geometric entities defined in the planning stage with respect to the CT reference frame. In this way the real operating cell is completely described in the simulation environment, where the patient, the target points, the trajectories, etc. are defined. Thus, the robot's movement to execute the planned tasks can be simulated and the simulation sequence is shown to the surgeon on a monitor located in the operating room; if the sequence is correct, the execution stage can be started.The transformation matrix is exact as long as the patient is not moved by any surgical procedure executed on him. Should this happen, it is necessary to recalibrate the system using three reference points located on the head holder, which is assumed to be fixed with respect to the patient's head. In this case too, the approximate position of the calibration points is measured using the three-dimensional digitizer so that the laser sensor can be positioned near them.

4. PROCEDURE FOR MEASUREMENT OF THE REAL SURFACE

In this work a technique for measuring the real surface of the patient's skull for intraoperative matching has been developed, which utilizes a laser sensor and a 6 d.o.f. industrial robot.The laser sensor uses a triangularization method allowing to measure its distance from a surface characterized by a diffuse reflection. The precision of the measurement is independent from the type of material constituting the surface. Furthermore, the system is totally non invasive for the patient.

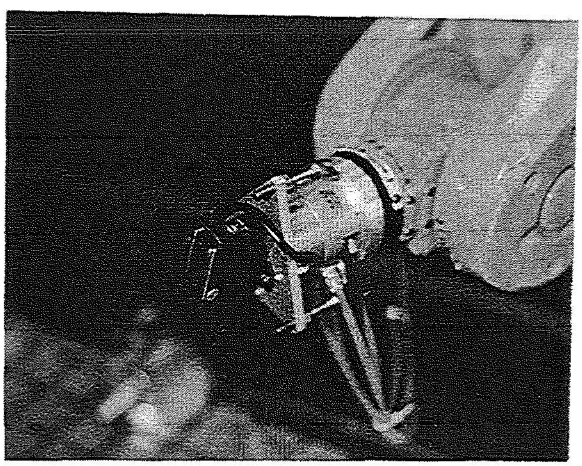


Figure 3 - The laser sensor mounted on the robot's end effector.

The sensor is characterized by a midrange distance about which there is a displacement range where the output signal vs. distance characteristic can be considered linear.The sensor is a Micro-Epsilon opto NCDT series 1605-100 whose main characteristics are:

Midrange	220 mm
Displacement range	± 50 mm
Non-linearity	300 mm

The sensor guarantees accurate readings if the measured surface presents a $\pm 15^{\circ}$angle with an axis of the sensor and $\pm 30^{\circ}$with an axis perpendicular to the former. This limit makes necessary to orient the sensor during the measurement of curve surfaces to guarantee as much as possible that the laser beam be perpendicular to the surface in the measurement point.Since the skull's surface does not have a defined shape, it is necessary to introduce a measurement procedure that adjusts the direction of the laser beam on the basis of the previous readings. Thus, the sensor has been mounted on the robot's end effector: in this way the sensor can be positioned and oriented in whatsoever direction in the whereabouts of the skull. This guarantees a repeatability of ± 0.1 mm and a precision of ± 0.1 mm. As it has been described above, the measurement procedure starts with a coarse calibration of the robot with respect to the skull in order to define the workspace.When the patient has been positioned on the operating bed, the approximate position of the skull is determined by reading the fiducial markers with the three-dimensional digitizer. This provides an approximate transformation matrix that allows to begin correctly and safely the measurement procedure.A midrange surface is generated in the virtual model of the skull; this surface has an average offset equal to the sensor's midrange. The midrange surface must be external to the bounding security volume, that is the volume that may not be entered by the robot.Then, the robot positions itself in that point of the midrange surface which corresponds approximately to the top of the skull, with a direction perpendicular to the modeled surface. Thus, the first measurements with a fair precision can be made, allowing a self calibration of the system, that will be now described. Three readings are made on the vertices of an equilateral triangle, described on the midrange surface in the whereabouts of the point that has been determined on the top of the skull. These three readings allow to calculate the normal to the real surface in the center of the same triangle. The measurement sequence proceeds with a series of points located along a spiral on the surface. At each further step the reading direction is determined on the basis of the last three measured points. These three points are chosen in such a way to guarantee a good approximation in the determination of the local normal. However, the reading procedure is not critical because the skull's surface is regular enough. The result of the measurement is a set of points describing the patient's head surface in the robot's reference system.

5. CONCLUSION

Surface matching is a suitable non-invasive registration method for multimodal correlation and intraoperative matching. It guarantee, with the minimum discomfort for the patient, registration errors compatible with the required precision. It can be successful applied to multimodal correlation of diagnostic exams in which reference surfaces can be extracted thru interactive segmentation programs. The same surfaces can be used for the intraoperative matching by means of a laser sensor mounted on the end effector of an industrial robot for the "real surface" detection.

REFERENCES

[1] S. Bodice, R. Caracciolo, M. Giovagnoni, A. Rossi, "An On-Line Adjusting System for Assembly Robots", Proc. of the Second International Symposium on Measurement and Control in Robotics, Tsukuba Science City, Japan, November 1992.
[2] N. T. S. Evans, "Combining Imaging Techniques", Clin. Physiol. Meas., vol. 11, Suppl. A, pagg. 97-102, 1990.
[3] R. Dann et al., "Evaluation of Elastic Matching System for Anatomic (CT, MR) and Functional (PET) Cerebral Images", Journal of Computer Assisted Tomography, 13(4): pagg. 603-611, July/August 1989.
[4] P. T. Fox, J. S. Perlmutter, and M. E. Raichle, " A Stereotactic Method of Anatomical Localization for Positron Emission Tomography", Journal of Computer Assisted Tomography, n.1, January/February 1985, Raven Press, New York.
[5] T. M. Peters, J. A. Clark, et al., "Integrated Stereotaxic Imaging with CT, MR Imaging, and Digital Subtraction Angiography", Radiology, 1986; 161: 821-826.
[6] D. J. Hawkes, D. L. G. Hill, et al.,"Preliminary Work on Interpretation of SPECT Images with the Aid of Registered MR Images and MR Derived 3D Neuro-anatomical Atlas", NATO ASI Series, Vol. F60, 3D Imaging in Medicine, 1990, Edited by K.H.Hohne et al.
[7] B. J. Erickson and C. R. Jack, "Correlation of Single Photon Emission CT with MR Image Data Using Fiduciary Markers", AJNR 14:713-720, May/June 1993.
[8] P. Clarysse, D. Gibon et al., "A Computer-Assisted System for 3-D Frameless Localization in stereotaxic MRI", IEEE Transactions on Medical Imaging, vol.10, n.4, December 1991.
[9] D. N. Levin, "MR and PET data merge in 3-D images of brain", Diagnostic Imaging International, 28:33, January/February 1990.
[10] C. A. Pelizzari, G. T. Y. Chen et al., "Accurate Three-Dimensional Registration of CT, PET, and/or MR Images of the brain", Journal of Computer Assisted Tomography, 13(1): 20-26, January/ February, 1989 Raven Press, Ltd., New York.
[11] C. A. Pelizzari, G. T. Y. Chen, "Image Correlation Techniques in Radiation Therapy Treatment Planning", Computerized Medical Imaging and Graphics, vol. 13, n. 3, 235:240, 1989, Printed in the U.S.A.
[12] B. L. Holman, R. E. Zimmerman et al., "Computer-Assisted Superimposition of Magnetic Resonance and High-Resolution Technetium-99m-HMPAO and Thallium-201 SPECT Images of the Brain", The Journal of Nuclear Medicine, vol.32, n.8, August 1991.
[13] D. N. Levin, C. A. Pelizzari et al., "Retrospective Geometric Correlation of MR, CT, and PET Images", Radiology 1988; 169: 817-823.
[14] W. Zhao and T. Y. Young, "A new approach to three-dimensional reconstruction with application to autoradiographic images", Computers in Medicine, vol. 13, n. 3, 1991.
[15] S. Lavallée, P. Cinquin et al., "IGOR: Image Guided Operating Robot. Metodology, medical applications, results", Innov. Tech. Biol. Med.,vol.13, No 4,1992.
[16] D. Glauser, P. Flury, "Conception of a Robot Dedicated to Neurosurgical Operations", Proc. of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1991.

VIRTUAL REALITY AND TELEROBOTICS APPLICATIONS OF AN ADDRESS RECALCULATION PIPELINE.

Matthew Regan and Ronald Pose.

Department of Computer Science, Monash University,
Clayton, Victoria 3168, Australia.
Email: {regan,rdp}@cs.monash.edu.au

ABSTRACT.

The technology developed at Monash University and described in this paper was designed to reduce latency to user interactions in immersive virtual reality environments. It is also ideally suited to telerobotic applications such as interaction with remote robotic manipulators in space or in deep sea operations. In such circumstances the significant latency in observed response to user stimulus which is due to communication delays, and the disturbing jerkiness due to low and unpredictable frame rates on compressed video user feedback or computationally limited virtual worlds, can be masked by our techniques. The user is provided with highly responsive visual feedback independent of communication or computational delays in providing physical video feedback or in rendering virtual world images. Virtual and physical environments can be combined seamlessly using these techniques.

INTRODUCTION.

The combination of Delayed Viewport Mapping [1] as implemented using an Address Recalculation Pipeline, image composition [2], and Prioritized Rendering [3] provides not only an order of magnitude reduction in image rendering required for interaction with a given virtual world, but a useful tool for all head mounted applications. A further important benefit is the graceful handling of inadequate computational capacity without sacrificing image resolution or latency to interaction.

An Address Recalculation Pipeline (ARP)[1] is a hardware implemented algorithm which performs delayed viewport orientation mapping. Using an ARP it is possible to orientate a computer generated virtual world with a user's head orientation after the scene has been rendered rather than before, as is the case with conventional virtual reality systems. This drastically reduces the computational component of the latency perceived by the user. Latency to user head rotations is essentially removed and latency to user translations may be significantly reduced with the use of image composition and priority rendering.

With image composition a scene is divided into several sections, each being allocated to a different rendering engine. Thus there are several rendering engines drawing different parts of a scene into different display memories in parallel. When displaying the scene, the images in all of the display memories are composed. All of the pixels spread across the display memories which correspond to a screen location in the head mounted display or other display device, are fetched simultaneously and the pixel with the smallest Z-value is displayed. Conventional systems can achieve almost linear speedup with multiple rendering engines [2].

With the viewport independence provided by the ARP in a head mounted display environment prioritized rendering [3] can be employed. With this scheme one can update the different display memories at different rates making it possible to render only those parts of the scene which change or which are most important to update quickly rather than the entire scene. Note that this is independent of interactive latency.

Experiments have shown that the speed up achieved with prioritized rendering can be significant. In a sample virtual world the number of objects redrawn at any update was reduced by on average 90% [3]. Thus for M rendering engines we achieve a speedup of 10 M.

The low latency achieved by these methods based around an ARP may be applied to a augmented reality and telerobotics applications. In a telerobotic environment the latency to user head rotations tends to be quite high. The mechanical delays involved in making a robotic head follow the motion of a user's head are a significant component however the additional two-way communications delay is also important.

Using an ARP it is possible to correct for the difference in orientation between the most up to date head tracking information and the orientation of the robotic head when the image was captured. As a result the user may see an old image which has been remapped to compensate for its invalid orientation. The user sees all images with the correct orientation at the update rate of the display device, typically 60Hz. The ARP does not need to be updated at the maximum rate or even a predictable rate. Image compression which generally leads to unpredictable frame rates may be used without annoying side-effects.

THE ADDRESS RECALCULATION PIPELINE.

An address recalculation pipeline is a hardware implemented algorithm which performs viewport orientation mapping after rendering. Rather than using a simple conventional counter for display memory access the addressing mechanism becomes quite complex and provides a correction for wide angle viewing lenses and user head orientation as pixels are fetched from display memory.

The user head orientation doesn't need to be known accurately until the first pixel of a frame is to be displayed on the output device. As a result the latency to user head rotations caused by computational delays is in the order of two microseconds. This latency is independent of scene complexity and renderer overload.

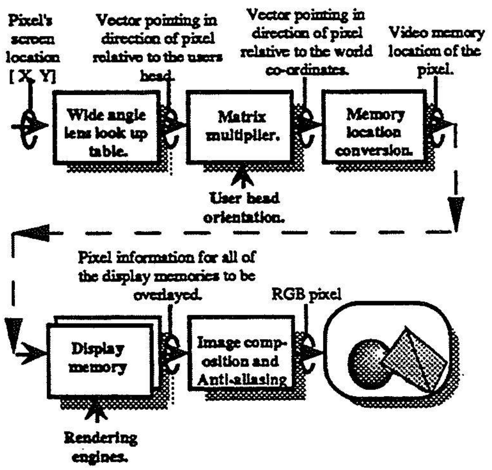


Figure 1. The address recalculation pipeline.

The pipeline is depicted in Figure 1. The first input to the pipeline is the X-Y screen location of the current pixel to be displayed. This screen location is provided by a conventional graphics controller at a normal pixel display rate. A look up table called a Wide Angle Viewing Lens Lookup Table converts the screen X-Y location into a 3-D unit vector pointing in the direction at which the pixel is seen by the user, through the wide angle lenses, relative to the user's head. The look up table has one entry per display pixel where each entry consists of three 16-bit vector components. For a display device of resolution 640 by 480 pixels the lookup table will require a memory of 1024(cols)*512(rows)*6(bytes per entry) = 3 Mbytes. Many head mounted displays use wide angle viewing lenses which preserve a standard viewport mapping, however if a special mapping is required for higher fields of view[6][7], the lookup table may be loaded with a new lens mapping, compensating for the lens mapping without run-time penalty.

The 48-bit output of the wide angle viewing lens feeds into a matrix multiplier which forms the next stage of the pipeline. The multiplier multiplies the pixel direction vector with a 3 by 3 matrix containing user head orientation information. The resulting output vector points in the direction at which the pixel is seen by the user, through the wide angle viewing lenses, relative to the world coordinate system. The pixel direction vector is fed into the matrix multiplier at pixel display rates while the head orientation matrix is updated at the start of each display frame (i.e.. after each vertical sync signal). The matrix multiplier is also implemented with 16-bit fixed point arithmetic and is built with nine commercially available 16-bit by 16-bit, 40ns multipliers and six 16-bit, 40ns adders. The output vector from the matrix multiplier is in the form of three 16-bit fixed point vector components [Vx Vy Vz].

The next pipeline stage, called the Vector Conversion stage, converts the 3D unit vector into a display memory location. The chosen display memory topology for this architecture is the surface of a cube. A spherical topology is also possible [5]. Figure 2 depicts an the face organization on the surface of a cube. The conversion process involves computing the point at which the ray intersects with the surface of a cube. When the cube is aligned to the axes of the coordinate system such that each face of the cube has one of its X, Y or Z coordinates fixed at +/- 1.0, the intersection may be computed with a set of parallel divisions with range checks on the outputs of the divisions. For example if the result of the divisions Vx/Vy and Vz/Vy are both within the range (-1.0, 1.0) the ray must intersect with only two of the six faces. The sign of Vy is then used to determine the face of intersection. The point of intersection on the face is then (Vx/Vy, Vz/Vy). The divisions must occur at pixel display rates, so the divisions are performed by a reciprocal lookup followed by a normal multiply using another set of 40ns multipliers. The reciprocal lookup has extra output bits which are used to compensate for classification with fixed precision arithmetic. A programmable logic device is used to accumulate data from the appropriate data paths to multiplex the divider outputs to form the display memory address.

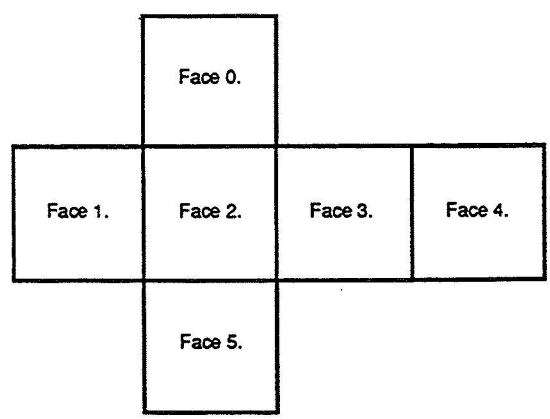


Figure 2. Display Memory Organization.

The address recalculation pipeline performs a remapping of the image in display memory to form an output image in real time based on the wide angle viewing lenses and the user head orientation. This remapping is performed one pixel at a time by the hardware in the address recalculation pipeline. The remapping occurs by sampling the image contained within the display memory and as with any form of discrete sampling, aliasing occurs. Even if the image in the display memory is anti-aliased and rendered with a high quality rendering technique, the hardware sampling occurring will cause aliasing in the final image. The aliases introduced by the address recalculation pipeline cannot be corrected with software in the rendering process. Any pipeline anti-aliasing must occur in hardware.

After simulating the possible artifacts caused by no hardware antialiasing strategy and considering the overall cost of an address recalculation pipeline system, hardware anti-aliasing is considered necessary. The anti-aliasing strategy chosen for this architecture is a linear interpolation filter using redundant addressing bits from the intersection computation. A linear interpolation filter provides an adequate trade-off between system expense and filter quality[8]. In order to perform linear interpolation the four pixels surrounding the point of intersection must be fetched simultaneously. The interleaving mechanism for fetching the four adjacent pixels and the method of interpolation are discussed in detail in [4].

VIRTUAL REALITY.

The ARP was designed specifically to compensate for many of the problems associated with HMD graphics systems. Image composition opens up a gateway into priority rendering which leads to significant gains in the effective rendering performance of a virtual reality graphics system.

Image overlaying or image composition [2] is a technique often used to increase the apparent display memory bandwidth as seen from the renderer. Rather than having one display memory (or two for double buffering) the graphics system has multiple display memories. Different sections of the visible scene may drawn into separate display memories then overlaid to form a final scene. In many implementations each display memory has a private rendering engine. The concept of image composition is depicted in figure 3.

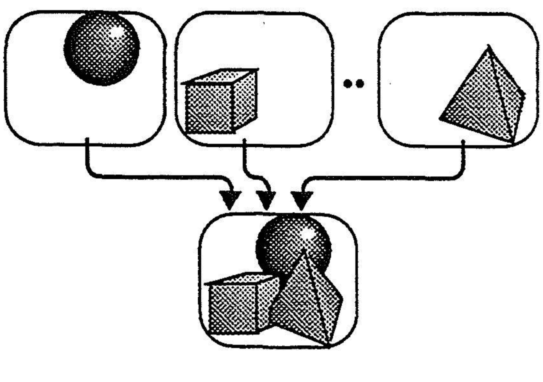


Figure 3. Image Composition.

Image composition allows the possibility of rendering different objects (down to a polygonal level) to different display memories. A side-effect of image composition is that each display memory may have its own unique update period. Using an ARP it is possible to make effective use of this side effect of image composition to achieve in certain cases much better performance increases from the available rendering hardware when compared to conventional image composition systems. This performance improvement eludes the conventional systems because the images in the display memories of a graphics system with an ARP do not necessarily become invalid when the user's head orientation changes, thus the length of time an image in display memory is valid only loosely depends on the orientation (for a stereo view). For example a non interactive background may never require re-rendering and may thus be pre-rendered with great detail using a high quality rendering technique and a complex model.

Using an ARP it is possible to render a scene which is largely independent of the user's head orientation. When image composition is combined with the address recalculation pipeline it is possible to render different parts of a scene at different rates this new paradigm is called priority rendering. Priority rendering is demand driven rendering. An object is not redrawn until its image within the display memory has changed by a predetermined threshold. In a conventional system this strategy would not be effective as almost any head rotations would cause considerable changes to the image in display memory and the system would have to re-render everything.

The threshold for determining when an object has changed by more than a tolerable amount is determined by the designer of the virtual world and may typically be based on several factors. Usually this threshold is in the form of an angle (θ_t) which defines the minimum feature size of the world. This value may vary from less than the minimum feature size the human eye can detect, to the size of one or more display pixels. Priority rendering attempts to keep the image in display memory accurate to within θ_t at the highest possible update rate.

In order to compute the period for which a given object is valid we compute the time it takes for the object to translate by θ_t, to grow or shrink by θ_t or to change by θ_t due to animation. The objects validity period may be computed from the size of the object, the distance to the object and the user's speed relative to that object. An additional factor is added by the designer of the virtual world which describes how much the object is animating. Once the period for which an object's image is valid has been determined the object may be assigned to a display memory with an appropriate update rate. A more detailed explanation of the computation of validity and assignment to display memories is given in [3]

The rendering hardware may have more display memories available than the virtual world requires for high efficiency. In this event, multiple renderers and display memories may be assigned to the one update rate thus devoting more hardware resources to a particular update rate, helping to balance the load.

Priority rendering may be used to reduce the overall rendering load on the rendering subsystem. The rendering load is based on several features of the scene, where the actual number of polygons is just one of the factors. One of our virtual world applications is a walk through of a forest. This simulation was performed in order to determine the rendering load on various display memories with various update rates.

In the experimental virtual environment the combination of the ARP, image composition and priority rendering cut the total number of objects requiring re-rendering by 90% when compared with the number of objects requiring re-rendering in an equivalent system without the ARP. That is, the system with the pipeline only had to redraw 10 objects for every 100 objects the system without the pipeline had to redraw for a similar illusion.

A stereo view of a virtual world is highly desirable within a head mounted graphics system. With an ARP the display memories are not actually centered around the point of rotation of the user's head, rather they are centered around the user's eyes. This means when the user's head rotates while the user is stationary a small amount of translation occurs. This implies the need to re-render some objects which are affected by the translation caused by the head rotation. Experiments have shown that head rotations smaller than 45 degrees require few objects to be updated due to the small translation of the eyes[3].

AUGMENTED REALITY.

An ARP graphics system with priority rendering may be used for augmented reality applications in the same way it is used for virtual reality applications, however the use of an ARP alone has significant advantages over a conventional graphics system when applied to augmented reality environment graphics systems.

The difference between virtual reality and augmented reality is in their treatment of the real world. Virtual reality environments immerse the user inside a virtual world that completely replaces the real world outside. In contrast augmented reality uses see-through HMDs that let the user see the real world and the virtual world at the same time. See-through HMDs augment the user's view of the real world by overlaying or compositing three-dimensional virtual objects with their real world counterparts. Ideally, it would seem to the user that the virtual and real objects co-exist.

Researcher in the field of augmented reality recognize that to use the technology in practice the 'registration problem' must be overcome[9]. The real and virtual objects must be aligned with respect to one and other, or the illusion that the two co-exist will be compromised.

The main sources of registration errors are,
- -Distortions in the HMD optics.
- -End-to-end system latency.
- -Mechanical misalignment in the HMD.
- -Errors in the head tracking system.
- -Incorrect viewing parameters (field of view, tracker-to-eye position and orientation, interpupillary distance)

Of these factors, only the first two factors may by improved by modifying the image generation process alone.

Distortions in the HMD optics in an augmented reality environment become particularly noticeable when the distortion of the image from the image generator does not match the view of the real world. Conventional real-time image generation systems tend not to provide a facility to correct for the distortions introduced by the optics in a HMD. If there is some form of distortion correction, it is usually prohibitively expensive.

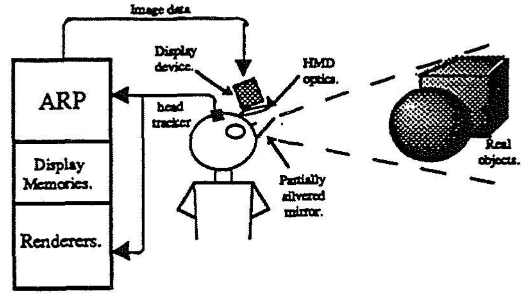


Figure 4. An augmented Reality environment.

An ARP provides a mechanism for correcting optical distortions introduced by optics within the HMD system. This mechanism derives from the versatility of the Wide Angle Viewing Lens Look Up Table (WAVELUT). The WAVELUT is a large lookup table which contains a direction vector for each pixel on the output display device. Provided the nature of the optical distortion of the HMD optics is known, the direction at which each pixel is seen by the user relative to the HMD may be computed. For each pixel in the displayed output an associated unit direction vector is computed and downloaded into the lookup table. The computation of this distortion is a one-off expense and allows for real-time correction of optical distortions without penalizing rendering performance. When new optics are installed in the HMD, or a new HMD is to be used, the optics need to be recomputed once. Such a setup is depicted in Figure 4.

A major feature of the ARP is that the update rate for user head rotations is bound to the update rate of the display device usually 60+ Hz, instead of the rendering frame rate. Also, with an ARP, the latency does not include the rendering time and doesn't include double buffer swap delays. The orientation of the view the user sees does not need to be known until the first pixel is to be sent to the display device. This means the images the user sees use the most up to date head tracking information. The nature of the latency to head rotations is depicted in Figure 5.

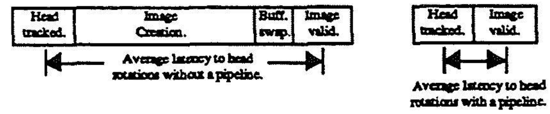


Figure 5. Latency to head rotations.

As a result end-to-end latency performance is improved when compared with conventional augment reality systems by by-passing the image generation component of the latency period for head rotations. The latency induced by the ARP is effectively less than two micro seconds and therefore may be neglected.

The magnitude of the end-to-end latency in an augmented reality environment is more critical than any other HMD application, such as VR or telerobotics[10]. This is because the user's visual system has real world objects which have zero latency to act as references against the virtual objects. An ARP is effectively capable of removing the latency for head rotations greatly improving the registration between the real and virtual objects.

The hardware in the ARP allows compensation for head orientation changes only, however priority rendering may be used to improve the performance of the system when user translations occur similar to the case of the conventional virtual reality environments.

TELEROBOTICS.

Telerobotics technology is a powerful way of allowing machines under human control to operate in environments that are hostile to humans. The goal of the technology is to convince a user that he or she is in the hostile environment to such a degree that the human user may perform complex operations through the robot, that a robot could not perform autonomously. Applications vary from robots in deep space building space stations to deep sea applications were robots repair and maintain underwater pipelines.

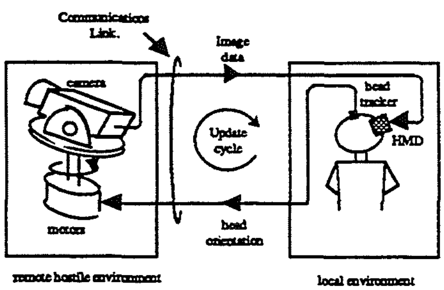


Figure 6. A conventional telerobotic setup.

Conventional telerobotic techniques usually feed the displays in an HMD from a set of cameras on a remote robotic head. The orientation of the robotic head is controlled with the information gained from the user head tracking equipment following the motion of the user's head. Ideally the motion of the robotic head would match the motion of the user's head. Figure 6 depicts a typical telerobotic setup with a HMD. End-to-end latency tends be exceptionally high in such a scenario. Once the user's head position has been tracked the orientation is sent via a potentially long communication path to motors controlling the orientation of the robotic head. These physical motors respond, move the robotic head to the desired position and capture an image. Next the image is sent back via the communications path to the HMD where it is displayed to the user. It becomes quite clear that even if very fast motors are used with short, high speed communications paths, the latency to orientation changes by the user's head will be very high.

Latency in a telerobotic environment is caused by a mismatch in the orientation of the users head and the orientation of the view the user sees in the HMD. The orientation of this view is the same as the orientation of the robotic head at some previous time. With an ARP it is possible to correct for the difference between the orientation of the robotic head when the image was captured and the current orientation of the user's head. Instead of feeding the image from the robotic head to the displays in the HMD, it is sent to the display memory of an ARP graphics system. At the start of each user update cycle, the users head orientation matrix is multiplied by an orientation matrix from the robotic head (which is obtained from sensors on the motors which move the robotic head) and represent the orientation of the real-world relative to the robotic head at the time when the image in the display memory was captured. The result is a matrix which converts the users head orientation to the robotic head coordinate system. This matrix is then fed into the ARP at the start of each user update. Such a setup is shown in Figure 7. As the display memory is divided into six faces, it is necessary to have multiple cameras on the robotic head to capture the entire image. The actual number of cameras required depends on the latency of the robotic update cycle however it is assumed that the number of cameras required is between four and six. This paper will not go into the physical details of the camera arrangement.

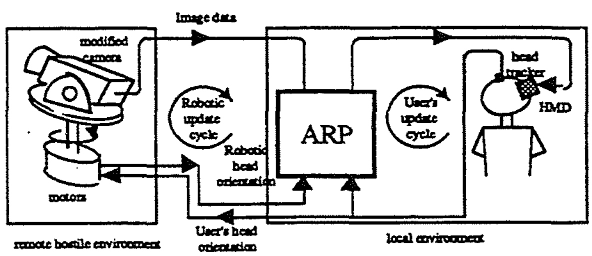


Figure 7. A telerobotic setup with an ARP.

The images in the systems display memory are valid for more than one update of the HMD, and as such it is possible to use the same image for a quite some period of time, this is similar to priority rendering in virtual reality where the user sees images that have been valid for some period of time. So while the orientation of the image the user sees is being updated at the frame rate of the output device, i.e. 60+Hz (user update rate), it is possible that the image stored in display memory is being updated at a lower rate, i.e. 5-30Hz (robotic update rate). No matter how long the communication delay to and from the telerobotic environment may be, the user always sees images with the correct orientation.

The update rate for the images coming from the robotic head need not match the update rate the user apparently sees. More importantly, the rate at which the images come from the robotic head need not even be predictable. As such the images from the robotic head may be compressed for transmission. The ARP will keep using the image last received until a new image has arrived. In a conventional system, such unpredictability of frame rate would be very noticeable especially when the user is looking at the most complex objects (hardest to compress). The use of image compression is extremely desirable when the communications path is long and expensive.

While the ARP does reduce the rotational latency, there will be latency to translations. To perform translations the robotic head actually needs to move, hence the latency will be bound to the robotic update cycle. However, latency to translations are common in everyday life, for example there is a delay between pressing on an accelerator in a car and motion of that car, such delays are easily tolerated by humans. Latency to head rotations are not seen in everyday life and may be very disorientating.

Applications which require a stereo view of the remote environment require 2 camera setups. Some latency to stereoscopy will be noticed as the user's head rotates, however latency to stereoscopy is easily tolerated by humans.

CONCLUSION.

In this paper we have described the how an ARP achieves low latency to head rotations in all HMD environment, and how it leads to an order of magnitude reduction in rendering costs, how it improves registration within an augmented reality system and how it may be used to hide often lengthy mechanical and communications delays in a telerobotic environment.

The low cost and high performance of an ARP makes it an ideal interface to a head mounted display. Whether the application is virtual reality, augmented reality or telerobotics the ARP has clear advantages over conventional systems.

ACKNOWLEDGMENTS.

Matthew Regan acknowledges the support of a priority Australian Postgraduate Research Award. This work has been supported by an Australian Research Council small research grant.

REFERENCES.

1. Regan, M. and Pose, R., "An Interactive Graphics Display Architecture". In proceedings of IEEE Virtual Reality Annual International Symposium, VRAIS 93, (Sept 18-23, 1993) Seattle WA. pp. 293-299

2. Molnar. S, "Image Composition Architectures for Real-Time Image Generation" Ph.D. Dissertation, University of North Carolina, 1991.

3. Regan, M. and Pose, R., "Priority Rendering with Virtual Reality Address Recalculation Pipeline". In proceedings of Siggraph 94. pp.155-162.

4. Regan, M. and Pose, R., "Display Memory Access Issues and Anti-Aliasing with a Virtual Reality Graphics Controller". Proceedings of the 9th Eurographics workshop on graphics hardware, (Sept 12-14, 1994), Oslo, Norway.

5. Regan, M. and Pose, R., "A Low Latency Virtual Reality Display System". Monash University, Department of Computer Science, Technical Report TR166, September 1992.

6. Robinett, W., and Roland, J., "A Computational Model for the Stereoscopic Optics for Head Mounted Display". In Presence 1, (winter 1992), pp. 45-62.

7. Deering, M., "High Resolution Virtual Reality", In proceedings of Siggraph 92, pp. 195-202.

8. Wolberg, G. "Digital Image Warping" IEEE Computer Society Press, 1990.

9 Azuma, R., and Bishop, G., "Improving Static and Dynamic Registration in an Optical See-through HMD", In proceedings of Siggraph 94, pp. 197-204.

10. Azuma, R., "Tracking Requirements for Augmented Reality.", CACM 36, 7 (July 1993), pp50-51.

1995109563

N95-15978

USING VIRTUAL REALITY FOR SCIENCE MISSION PLANNING: A MARS PATHFINDER CASE

351286

p. 10

Jacqueline H. Kim, Richard J. Weidner, Allan L. Sacks
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109

ABSTRACT

NASA's Mars Pathfinder Project requires a Ground Data System (GDS) that supports both an engineering and a science payload with reduced mission operations staffing, and short planning schedules. Also, successful surface operation of the lander camera requires efficient mission planning and accurate pointing of the camera.

To meet these challenges, the GDS Team designed a new software strategy that integrates virtual reality technology with existing JPL Navigational Ancillary Information Facilities (NAIF) and image processing capabilities. The result is an interactive, workstation-based application software that provides a high resolution, 3-dimensional, stereo display of Mars as if it were viewed through the lander camera. The design, implementation strategy and parameter specification phases for the development of this software have already been completed, and the prototype has been tested. When completed, this software will allow science investigators and mission planners to access simulated and actual scenes of Mars' surface. The perspective from the lander camera will enable scientists to plan activities more accurately and completely. The application also will support the sequence and command generation process, and will allow testing and verification of camera-pointing commands via simulation of the sequence.

This paper describes the architecture and characteristics of this science mission planning software now under development for Mars Pathfinder, including output from the prototype. Also, it addresses possible uses of this software by other planetary missions.

INTRODUCTION

The Mars Pathfinder Project is the first of NASA's Discovery Program missions. Discovery-class means low development cost ($150 Million or less), short development time (3 years or less), and focused science objectives. The Mars Pathfinder mission is the first lander mission since Viking in 1976. During the Viking Mission, many of the mission operations activities were labor-intensive, costly and time consuming, especially in the area of science mission planning for the imaging system. The Pathfinder Ground Data System (GDS) Team provided the functional design for a new software that eliminates much of the labor intensive work in mission planning for the lander imaging system, Imager for Mars Pathfinder (IMP).

The resulting software, named SIMP (Simulator for IMP), creates a "virtual Mars environment" on a workstation using high resolution, 3-dimensional, stereographic display of Mars terrain and atmosphere. This innovative use of workstation-based virtual reality enables scientists and mission operation's staff to plan the observations easily and accurately. This application supports the sequence and command generation process, as well as verification of the generated sequence.

The first prototype of the SIMP was tested recently and received favorable reviews by scientists as well as mission operations team. The prototype will go through several tests and refinement within the next 12 months. The SIMP can also be used for other future lander missions, such as landers in the Mars Program, with little modification.

MISSION DESCRIPTION

The Mars Pathfinder project development began in October 1993.

The spacecraft will be launched during the 1996 Mars opportunity (between December 5, 1996 and January 3, 1997), on a Delta II launch vehicle. The spacecraft will spend 6 to 7 months in cruise using a type I trajectory, and will land on Mars on July 4, 1997. The surface mission on Mars will be completed by August 1997.

The Mars Pathfinder mission's primary objective is to demonstrate the low cost engineering technology involving the cruise, entry, descent, and landing system required to place a payload on the Martian surface in an operational configuration. In addition, the lander carries a micro-rover as a technology instrument. The lander deploys the rover upon opening of the solar panels. The rover will be driven off of the solar panel after deployment of the solar panel.

The flight system consists of four main parts: 1) Aeroshell, parachute, and airbag Entry, Descent, Landing (EDL) System (See Figure 1), 2) Self righting, tetrahedral lander (See Figure 2), 3)Active thermal system for the lander (See Figure 3), and 4)Free ranging rover.

Figure 1 illustrates the EDL sequence. The spacecraft is enclosed within an aeroshell during cruise. After entering into the Martian atmosphere, a parachute is deployed to reduce the impact speed. Just before the landing, air bags are inflated to cushion the impact, and the parachute is jettisoned away from the lander position.

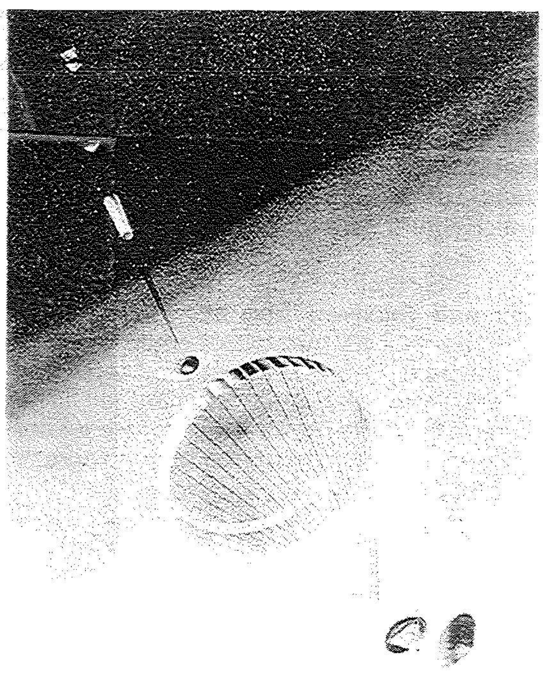


Figure 1 *- Mars Pathfinder Entry, Descent, Landing (EDL) Sequence*

The lander carries a significant science payload. There are several science instruments relating to atmospherics, meteorology, geology and imaging. Imager for Mars Pathfinder is the main science instrument, and it consists of a stereo camera with color image capability. The imaging system is located near the center of the lander and is controlled by a set of motors. Figure 3 shows the lander and its payload.

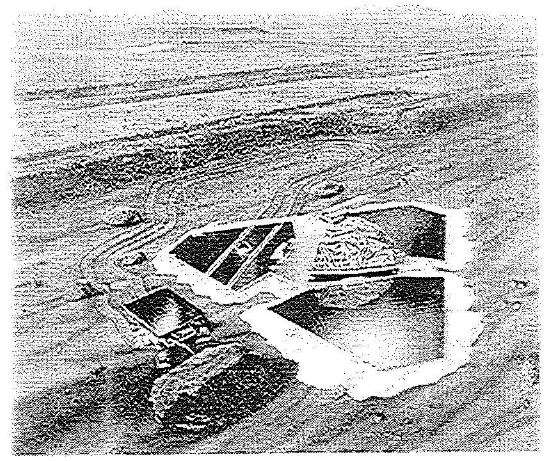


Figure 2 *- Artist's Conceptualization of the Lander on the Surface of Mars*

MISSION PLANNING FOR VIKING AND MARS PATHFINDER CAMERA

In order for scientists to control the camera, they need to specify approximately 32 different parameters, ranging from ephemeris information to optimum data compression ratio. The most basic parameters are displacement angle in azimuth and elevation direction (i.e. targeting parameters), and the location of the Sun.

During Viking mission operations, image mission planning was achieved by using the "Skyline drawings", timeline, and many hours of intensive calculations of essential quantities. Although this system served its purpose well, this was a labor-intensive, time consuming and inefficient process.

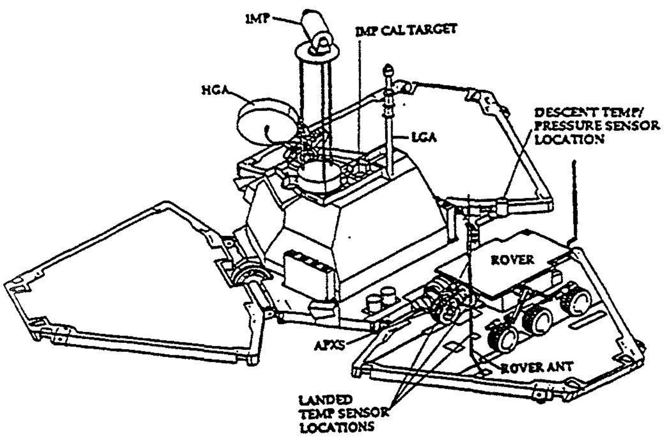


***Figure 3** - Pathfinder Lander Configuration*

Figure 4 shows the actual Skyline Drawings used by the Viking Lander-1, during Sol-0 and Sol-1. (A "Martian sol" is a solar day for Mars which is equivalent to 24.66 Earth solar hours.) The Skyline Drawings show image outlines on a rectilinear grid whose horizontal axis represents the azimuth and the vertical axis represents the elevation.

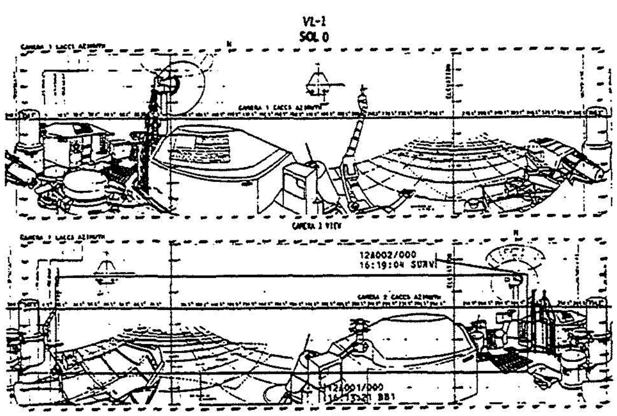


***Figure 4** - Skyline Drawings Used in the Viking Lander Mission*

The Viking Lander Timeline, shown in Figure 5, was also used as a main part of mission planning. This timeline shows the Viking Lander-1's mission activities during Sol 22. As one can see, these resources are unsuitable to use for new missions like Mars Pathfinder, considering advanced technology that is available today.

The proposal for the image mission planning tool for the Pathfinder was to create a 'virtual Mars' environment. The idea is to simulate the camera using a workstation such as Silicon Graphics or Sun. The simulator creates a virtual Mars environment on a workstation screen, displaying 3-dimensional, stereographic images of Mars' terrain as well as its atmosphere. The scene created should look as if it were seen by the camera. This idea utilizes existing hardware and entails only a small cost. The scenes of Mars will be viewed with stereo viewing devices already widely available.

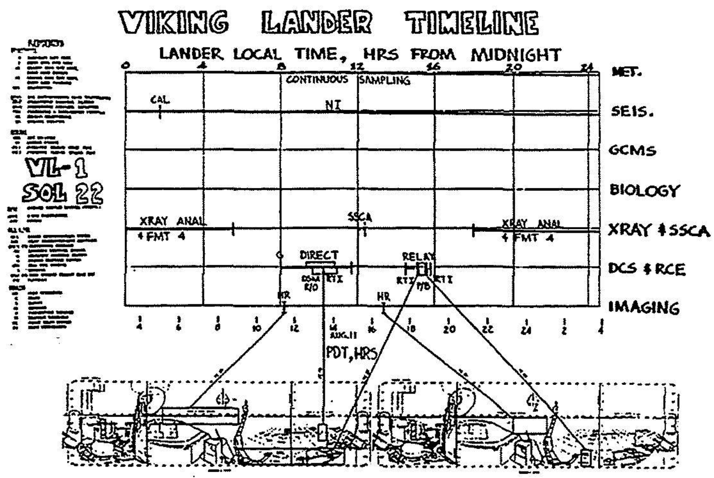


***Figure 5** - Viking Lander Timeline*

DESIGN OF SIMP

The SIMP design approach was different from the traditional approach at JPL. The decision was made to start the design without a detailed functional requirement document or schedule. Instead, this software development relied on close interaction between team members. This approach allows all team members to understand the purpose of the task, and the significance of their roles toward accomplishing the task. Most importantly, each member is completely responsible for his contribution to this task, but at the same time, to work as a cooperative team member.

The design phase of development began with a meeting between representatives of the Ground Systems, Image Processing, Science, and SIMP implementation team. During this initial meeting, the purpose and functions were identified, and concepts for creating a virtual Mars were proposed. In addition, the description of the mission, launch schedule and mission operations plan were discussed. The design team agreed that providing a comprehensive user's guide and a software description would be the only necessary documentation. The design team also decided that incorporating NAIF/SPICE is an appropriate approach to get ancillary information. (The description of NAIF/SPICE is attached as Appendix A.)

After a few months, the initial prototype was completed. The design team presented the prototype to scientists for a review, and found the prototype to be satisfactory in terms of meeting a foreseen necessary functions for mission planning. At this point, the implementation team decided to proceed with detailed development of SIMP, including some additional requested functionality. Figure 6 shows the prototype of SIMP.

SIMP creates a virtual Mars environment by displaying either single or mosaic images, in stereo view. There is an option to display either the Mars local coordinate reference or lander center coordinate reference on the border of the scene. Graphical display of azimuth and elevation angle, field of view and field of regard (ranges within the camera can move) are placed below the scene. In addition, there are text display of other parameters such filter and exposure information. On the scene, motor step grid and field of view are overlaid for quick reference.

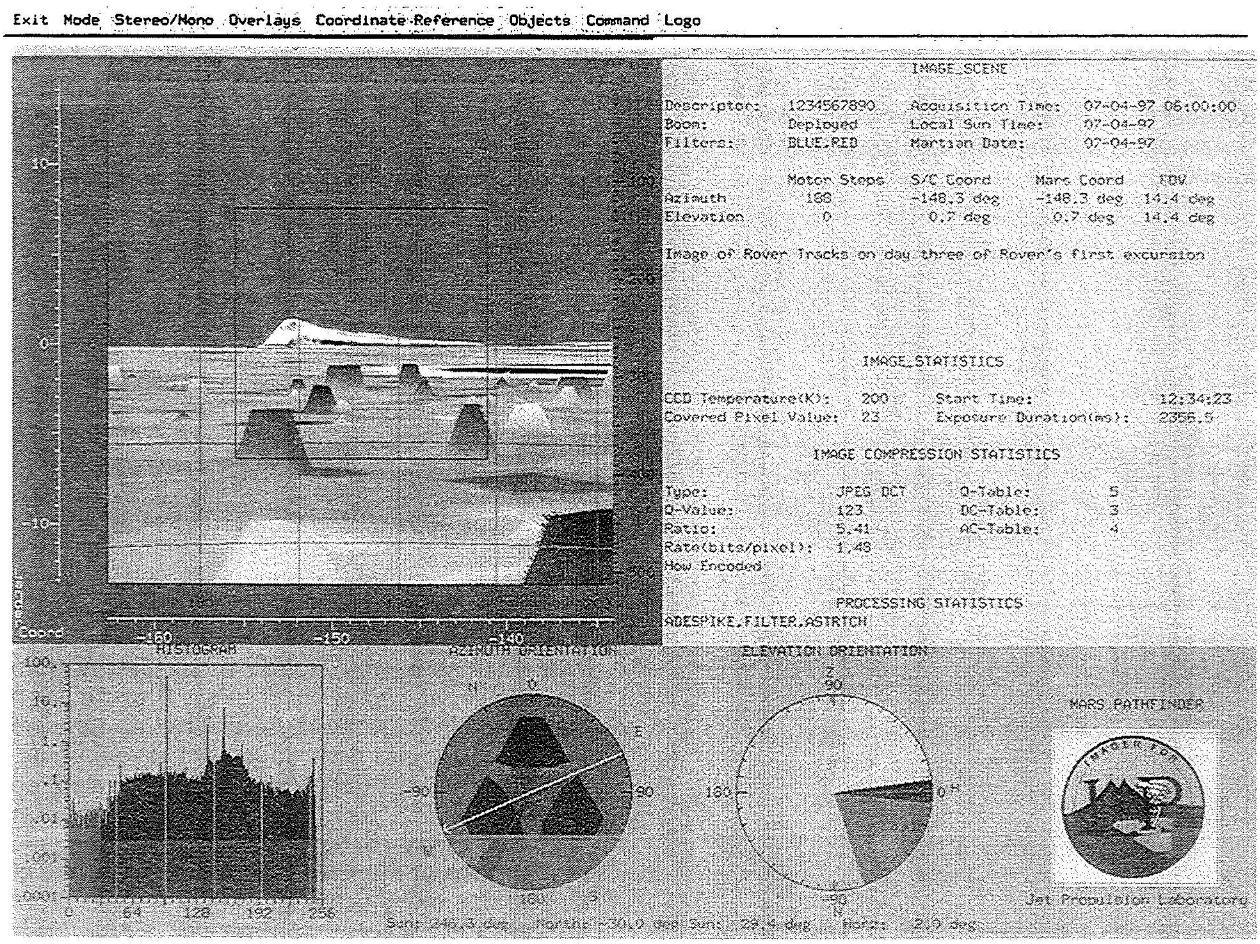


Figure 6 - Display from SIMP Prototype

PLAN FOR TESTING AND REFINEMENT

It is our plan to refine SIMP through continuous communication between GDS, the implementation team and the science team. Also, SIMP is schedule to be connected to the camera engineering model to test the real-time interface and its accuracy.

Within the next month, the design team of SIMP will participate in a special test activity, in cooperation with the science team. The Principal Investigator of the camera at the University of Arizona will create a pseudo-Mars, named Mars Garden, by designing an area according to available Mars data. The engineering model of IMP will be mounted within the Mars Garden and will be connected to a control system. At that time, SIMP will be connected to the control system to simulate the mission operations environment. This test will give us useful accuracy data as well as usability information about SIMP.

Figure 7 is a conceptual model of Mars Pathfinder's uplink (command) system. SEQGEN and SEQTRAN are sequence processing software currently used at JPL. SASF and SSF are input files for SEQGEN and SEQTRAN. As shown, SIMP will provide planning function as well as validation of the designed sequence. During next phase of Mars Pathfinder's system testing, SIMP will be placed within the uplink system to be integrated with the rest of the GDS.

POTENTIAL USE BY FUTURE LANDER MISSION

Currently, there is a proposed plan to repeatedly launch landers as Mars to conduct various scientific investigations. SIMP can be used to simulate any camera or remote sensing instrument on any of the future landers, with little modification. The only foreseeable modification required will be to generate SPICE kernel files for spacecraft and the camera. SIMP is independent of landing location or mission dates.

SUMMARY

Through design and development of SIMP, the Mars Pathfinder GDS has shown that an innovative use of virtual reality concept produced a high quality, re-usable tool for science mission planning. The resulting tool will accommodate scientists with accurate information when planning or validating their mission sequence. Furthermore, this tool can be used for any future landers regardless of time or the destination of the mission.

Note:
The SIMP tool development is part of the Model Based Planetary Tools Analysis Task funded by Joe Bredenkamp of Code ST, NASA. This effort described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology under contract with NASA.

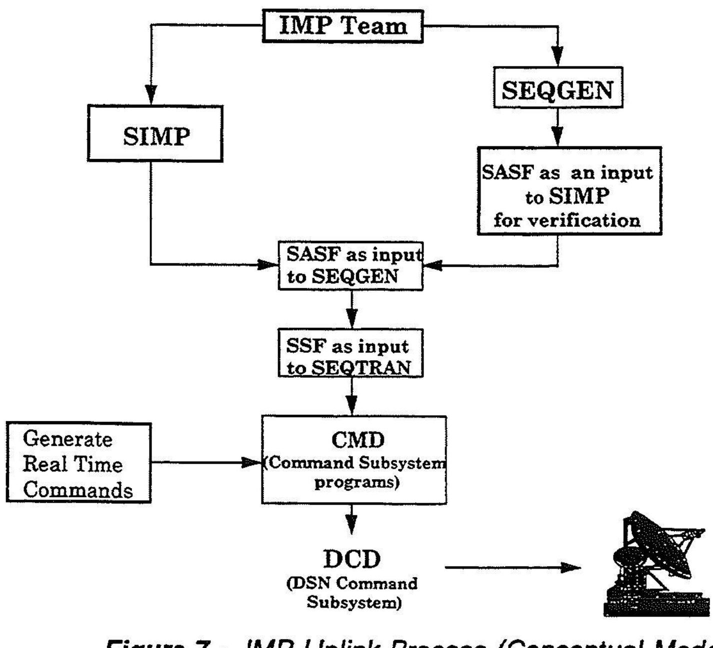


Figure 7 - IMP Uplink Process (Conceptual Model)

APPENDIX A

"Kernel Knowledge"
NAIF/SPICE Description, published by NAIF Group at JPL. (attached)

BIBLIOGRAPHY

Pathfinder Project Mission Plan (JPL D-11355), Preliminary version, 1993.

Viking Lander Imaging Investigation: Picture catalog of Primary Mission and Experiment Data Record (NASA Reference Publication 1007) by Robert B. Tucker, published by NASA, 1978.

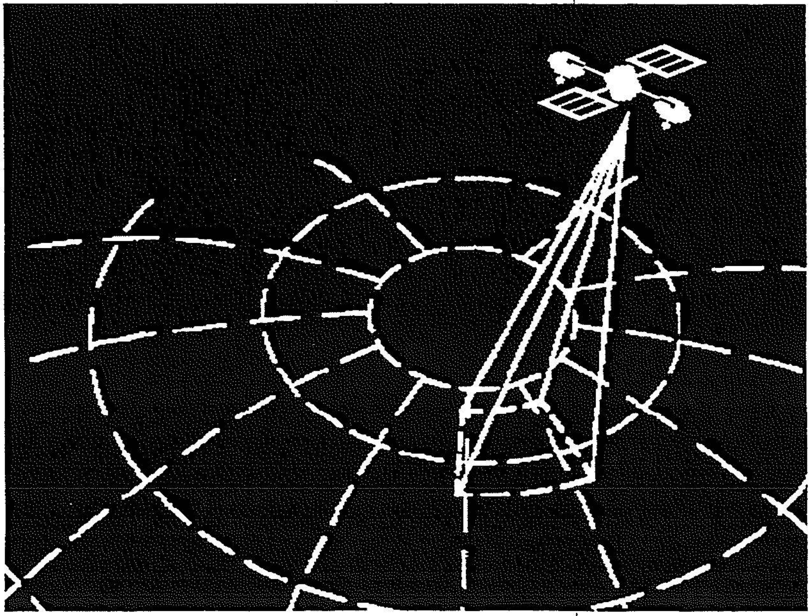
Mars Environmental Survey: Pathfinder and Network, information material published by NASA/JPL, 1993.

Mars Pathfinder Fact Sheet, information material published by NASA/JPL, 1994.

Kernel Knowledge, information material published by NAIF/SPICE Group at JPL.

Kernel Knowledge

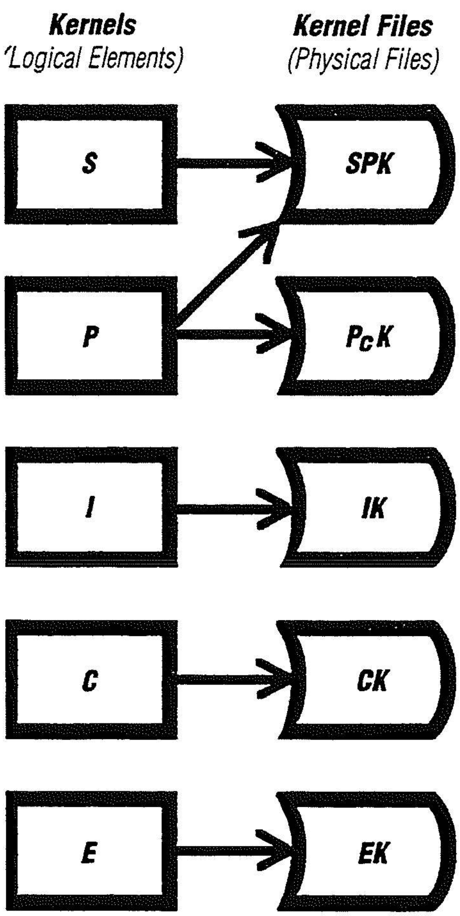
Navigation
Ancillary
Information
Facility



Mapping SPICE Kernels Into Real Products

Historically, the ancillary data needed to support the planning and analysis of observations made by instruments on spacecraft have been organized into five logical elements—called *kernels*—as shown in the left half of the figure, and as described in the table. The acronym SPICE was coined to refer to these kernels.

Kernel	Contents	Description
S	*Spacecraft ephemeris*	*Position and velocity of a spacecraft as a function of time.*
P	*Planet ephemeris and constants*	*Position and velocity of a planet, satellite, comet, asteroid, or the Sun as a function of time. Also, cartographic constants for that object.*
I	*Instrument descriptions*	*Instrument mounting alignment, internal timing, and other information needed to interpret measurements made with an instrument.*
C	*Camera pointing*	*The inertially referenced attitude (pointing) for an instrument or other spacecraft structure as a function of time.*
E	*Events*	*Spacecraft and instrument commands, ground data system event logs, and experimenter's "notebook" records.*



Kernel data are distributed in five kinds of files, called *kernel files*, as listed in the table (right.)

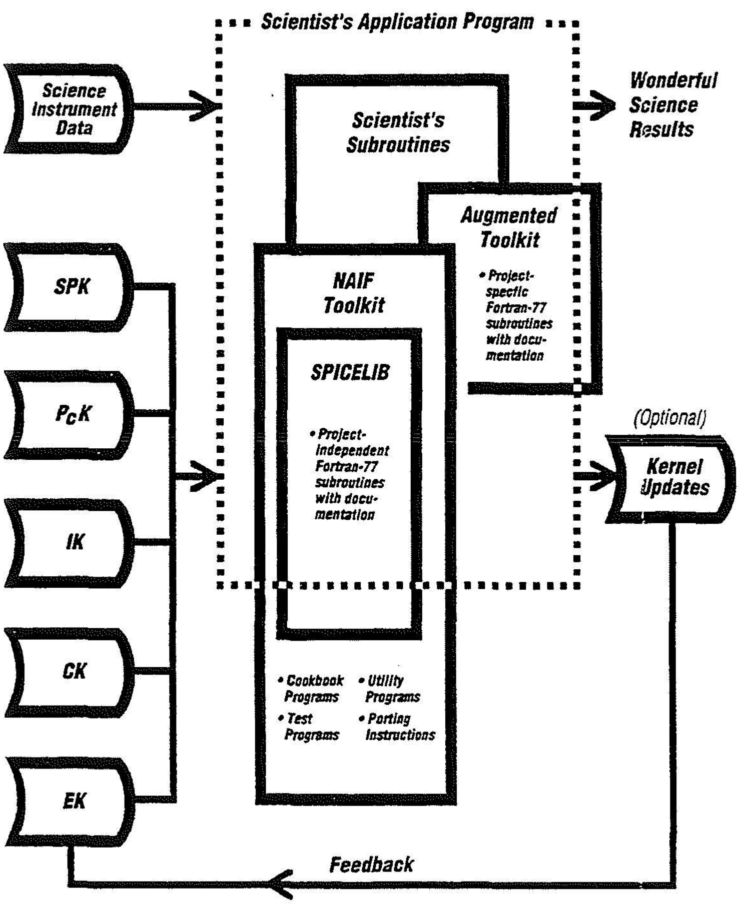
File	Format	Description
SPK	*Binary*	*Data from the S kernel, or from the ephemeris portion ($P_{ephemeris}$) of the P kernel, or both.*
P_CK	*Text*	*Data from the constants portion ($P_{constants}$) of the P kernel.*
IK	*Text*	*Data from the I kernel.*
CK	*Binary*	*Data from the C kernel.*
EK	*Text*	*Data from the E kernel.*

Data from the S and the $P_{ephemeris}$ kernels are generally used together (the state of a spacecraft is normally defined with respect to a planetary object), and may be included in a single file.

Most kernel files are originally produced by a flight project, such as Magellan, Galileo, or Mars Observer. Updates to these files could be produced both by the project and by planetary scientists during the course of their analyses.

In addition, NAIF produces some SPK kernel files containing planet, satellite, comet, asteroid, and Sun ephemerides for general use. These kernel files are derived from reference ephemerides provided by the Jet Propulsion Laboratory's Navigation Systems Section.

Using the SPICE System



SPICE Helps Interpret Science Instrument Data

The elements of the SPICE system—kernel files and software—are used to support the planning and analysis of space science data.

A scientist's application program might use pictures from the Mars Observer Camera to help determine the suitability of a particular region as a landing site for a sample return mission. The primary scientific result would be estimates of parameters that define local topography. A secondary result could be improved estimates of precisely where the camera was pointed when the pictures were taken; these data could be placed in a new C-kernel file.

SPICELIB—the principal SPICE software—is a collection of subroutines written in ANSI Fortran-77. Some of these subroutines read, write, and port binary kernel files, and read text kernel files. (Binary files are converted to an intermediate text format for transfer between various computers.)

The remaining subroutines use the information contained in those files to compute the geometric quantities (vectors, angles, distances) needed to plan observations or to interpret the data returned from science instruments. Each subroutine includes the information needed to select and properly integrate the subroutine into the scientist's own application programs.

The NAIF Toolkit consists of SPICELIB source code, including documentation, plus the following additional items:

Cookbook Programs are highly annotated, working programs that illustrate how SPICE kernels and SPICELIB subroutines may be used to compute commonly requested geometric quantities. (Sample kernel files are included.)

Test Programs can be executed by a Toolkit recipient to verify that the Toolkit code has been successfully ported to the recipient's own computer.

Utility Programs fall into two categories. Some can be used to examine and convert binary kernel files; others can be used to gain easy access to descriptions of SPICELIB subroutines.

Porting Instructions identify changes that nee to be made when the Toolkit is moved between various computers.

SPICE Offers Wide Applicability

The NAIF Toolkit can be used in planetary, space physics, and Earth scienc applications. It may be augmented with project-specific subroutines as needed, with NAIF normally providing this code under project funding.

Scientists or engineers pick needed Toolkit subroutines and combine them with their own software to create an application program. (Toolkit users should *not* revise Toolkit subroutines.)

When used in support of NASA flight project, SPICE kernel files and Toolkit software (including augmentations), are archived with science instrument datasets for future reference.

Navigation Ancillary Information Facility
Jet Propulsion Laboratory
Mail Stop 301-125L
4800 Oak Grove Drive
Pasadena, California 91109

What is SPICE ?

SPICE is a NASA information system for assembling, archiving, distributing, and accessing geometric and related ancillary information used to plan space science observations and interpret space science instrument data. This brochure describes the content and use of the basic SPICE system components.

The SPICE concept was defined by planetary scientists, and is being implemented by the staff of the Navigation Ancillary Information Facility (NAIF) at the Jet Propulsion Laboratory, with oversight by the science community.

Funding for development of the SPICE system is provided by the Information Systems Branch and the Solar System Exploration Division of NASA's Office of Space Science and Applications.

Coordinated Control of a Dual-Arm Dexterous Robot Using Full Immersion Telepresence and Virtual Reality

Larry Li [1], Brian Cox [2], Susan Shelton [2], Myron Diftler [2]

[1] NASA Johnson Space Center
[2] Lockheed Engineering and Sciences Company
Houston, Texas 77058
713-483-9160(voice) 713-483-7580(fax)
li@mickey.jsc.nasa.gov

ABSTRACT

Telepresence is an approach to teleoperation that provides egocentric, intuitive interactions between an operator and a remote environment. This approach takes advantage of the natural cognitive and sensory-motor skills of an on-board crew and effectively transfers them to a slave robot. A dual-arm dexterous robot operating under telepresence control has been developed and initial evaluations of the system performing candidate EVA, IVA and planetary geological tasks were conducted. The results of our evaluation showed that telepresence control is very effective in transferring the operator's skills to the slave robot. However, the results also showed that, due to the kinematic and dynamicsinconsistencies between the operator and the robot, a limited amount of intelligent automation is also required to carry out some of the tasks. Therefore, several enhancements have been made to the original system to increase the automated capabilities of the control system without losing the benefits of telepresence.

KEYWORDS AND PHRASES

Anthropomorphic, dexterous robotics, human-factors, telepresence, virtual reality.

INTRODUCTION

The current baseline approaches to robot teleoperation in the Space Shuttle as well as on the International Space Station Alpha (ISSA) are based on "joystick" type hand controllers. The visual feedback is provided by multiple cameras, most of which are mounted on the robot arms and at the worksite. For demanding tasks that require a high degree of coordination, the "joystick" approach is inadequate, and may overload the visual and manual capacities of the operator. As a result, the operator's skill is not effectively transferred to the slave robot. A different approach to robot teleoperation is *telepresence*. In telepresence, the master control and feedback devices are designed to maximize the use of the operator's innate cognitive and sensory-motor skills [1][3].

This paper describes an evolving telerobotics testbed at the NASA Johnson Space Center (JSC) that utilizes virtual reality (VR) and telepresence as its baseline mode of operation. The testbed consists of a master and a slave system. The slave system is a dual-arm dexterous robot called the Dexterous Anthropomorphic Robotic Testbed (DART). DART is controlled by the Full Immersion Telepresence Testbed (FITT), which is the master system of the overall testbed. FITT consists of several VR related input and output devices including a speech recognition system[3][4].

Besides describing the overall system, this paper will also discuss the results of our preliminary evaluations and the enhancements made to improve the capability of the original system.

OBJECTIVES

The main objective of the DART/FITT testbed is to develop and demonstrate technologies leading to a highly versatile and productive space telerobot. Specific objectives include: (1) develop a control scheme that permits the human operator to easily coordinate complex robot motions in demanding space tasks; (2) improve versatility and productivity; (3) maintain compatibility with existing and future crew interfaces (e.g., handles, tools); and (4) build in the capability for the testbed system to evolve.

DESIGN APPROACHES

Due to the inability of today's autonomous robots to perform complex unplanned tasks in a non-stationary, unstructured environment, we chose teleoperation as the baseline mode of operation. In teleoperation, the human operator can provide the cognitive and sensory-motor skills necessary to carry out these tasks.

With teleoperation chosen as the baseline mode of operation, the challenge now becomes how teleoperation can be made more effective. To meet this challenge, we applied the telepresence and VR technologies.

To complement an ergonomic telepresence/VR interface, we designed the slave robot to take on a human-like configuration and dexterity, so that the master-to-slave mapping is more direct.

Finally, we developed an open control architecture for shared control and "plug-and-play" capability. Shared control can increase the robot's productivity through the use of automation without sacrificing the versatility offered by the human operator. The "plug-and-play" modularity of the architecture will permit the robot to evolve by incorporating new automation capabilities as they emerge.

Following the above approaches, we developed the DART/FITT testbed system for laboratory evaluation (Figure 1). The following section describes the DART/FITT system in greater detail.

FITT

The FITT testbed, shown in Figure 2, is centered around a motorized chair and includes equipment for controlling DART's head camera unit, robotic arms and hands. The FITT also includes foot pedals that command direct drive motors on both the FITT base and the DART base, as well as initiate and terminate voice commands.

A VR helmet displays the remote stereo camera images with a 60 degree field of view and includes stereo head phones for audio feedback and a microphone for voice commands. The depth perception provided with stereo imaging is one of the testbed's most important immersion features. A magnetic tracker sensor located on the top of the helmet commands the orientation of the remote robot's camera head unit. The same sensors attached to each of the operators wrists provide x,y,z and roll, pitch, yaw controls for manipulator tool points. Glove controllers worn by the operator read the finger joint angles and use the information to control the robotic hands.

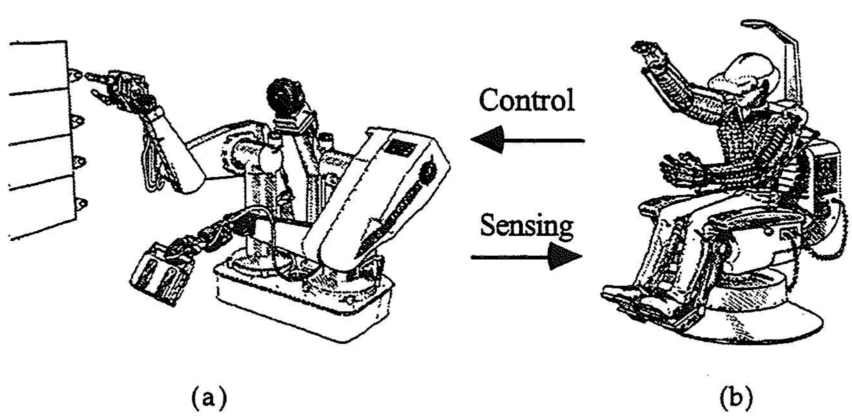


Figure 1. Telepresence control of a dual-arm dexterous robot. (a) The Dexterous Anthropomorphic Robotic Testbed. (b) The Full Immersion Telepresence Testbed (concept drawing).

Since the operator's hands and eyes are virtually immersed in the robot's environment, and are not available for initiating keyboard commands, the voice recognition system provides a convenient means of blending automated commands with the baseline telepresence control. The operator simply presses a foot pedal, gives a verbal command, and then releases the pedal. This technique prevents the voice recognition system from picking up extraneous inputs. The command is also processed and played back to the operator over a voice synthesizer for confirmation. These voice commands vary in complexity from a simple repositioning of a robot arm relative to a human arm (to take advantage of the greater travel of the robot arm) to a more complex maneuver such as grappling onto a dial and turning it a preprogrammed number of times.

The software that communicates and controls the FITT systems is hosted on a UNIX/VME workstation and a 486 PC equipped with a voice recognition board. Data from the magnetic tracker sensors, the glove controllers, and foot pedals, is sampled at approximately 100 Hz and then sent out over Ethernet using the TeleRobotics Interconnect Protocol (TelRIP)[7].

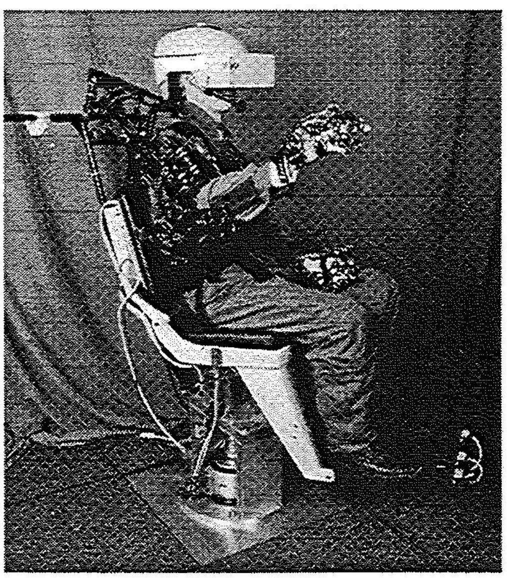


Figure 2. Full Immersion Telepresence Testbed.

TelRIP is a high-level, object-oriented communication package that makes the low level socket interfaces transparent to the programmer. For example, the voice recognition system samples data at a natural speaking speed and sends out TelRIP objects that initiate a prescribed semi-automated or automated action. A remote robot such as DART can set up its client communications program to receive any or all of the commands from FITT by registering "interest" in the appropriate TelRIP objects.

The force-reflective Exoskeleton Arm Master (EAM), worn by the operator in Figure 2, is not currently integrated with the FITT system. Nevertheless, we expect that the operator will be able to perform more complex tasks with an increased level of performance once the EAM is integrated with the FITT system.

DART

DART, shown in Figure 3, includes several robotic devices, controllers, and supporting workstations. The robotic arms are PUMA 562's each with an 8.8 pound payload capability. Each arm also has a force-torque sensor. On the right arm is a Stanford/JPL hand. Each finger has a urethane fingertip to provide a high static friction surface and can be hyper-extended to provide a large manipulation envelope. On the left arm is a parallel jaw gripper. The head camera unit that provides video feedback to the teleoperator supports 3 axes of rotations and contains two color CCD cameras. The driver level software is executed on two Tadpole™ multiprocessor systems. Each multiprocessor system has four M88000 processors and runs a multiprocessor version of the UNIX operating system. The vision system is implemented on a DataCube™ pipeline image processor board.

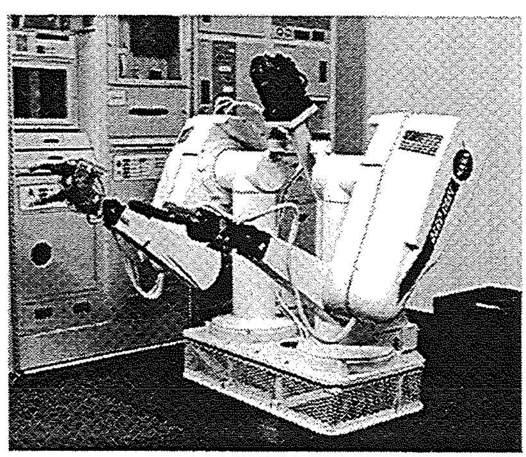


Figure 3. The Dexterous Anthropomorphic Robotic Testbed (DART).

CONTROL ARCHITECTURE

DART and FITT each has a distributed control architecture. Each subsystem spans one or more processes. The subsystem processes are distributed across several different computers, networked on an Ethernet backbone. Figure 4 shows how the DART and FITT systems are networked. Computers to the left of the SPARC-10 are part of the DART testbed; computers to the right of the SPARC-10 are part of the FITT testbed. The SPARC-10 itself serves as a message router and hosts the speech synthesis software.

The software architecture is shown in Figure 5. The subsystem processes communicate and are synchronized by TelRIP. This architecture provides a flexible environment for development, maintenance, and future enhancements. FITT controls DART by linking to this Ethernet backbone and commanding the subsystems through TelRIP. The router process, denoted by **R**, is responsible for transmitting data to the appropriate subsystem processes.

PRELIMINARY EVALUATIONS

Preliminary evaluations of the DART/FITT system were conducted using operators of varying skill levels, ranging from several years of robotic experience to absolutely no engineering experience. This allows the intuitiveness of operation to be qualitatively evaluated. The tasks ranged from inspection to object handling to dexterous manipulation.

Inspection tasks were comprised mainly of bringing an object towards the head camera and viewing it from different angles. These tasks provide information about the required display resolution, stereo perception, as well as the effect of working with *egocentric* views of the workspace. The object handling tasks include picking up objects of various sizes and shapes (*e.g.*, balls, pipes, tools) and placing them at a different location, and handing objects back and forth between the dexterous hand and the gripper. Some of the dual-hand dexterous tasks performed were tying a knot with a rope, folding and unfolding a thermal blanket, and manipulating an electronic task panel which contains toggle and rocker switches, push buttons, sliders, and a dial. These tasks reflect some of the basic dexterity and skills required for on-orbit extra- and intra-vehicular activities (EVA/IVA).

To further evaluate the DART/FITT system as a "planetary geologist", we put the system through a battary of tasks including holding up a light source while the operator examined a rock sample, picking up a rock sample and placing it into a bag or container, chipping a boulder with a hammer, picking up rock samples with an extended tong, and placing a gnomen next to a rock sample as a scale and color reference.

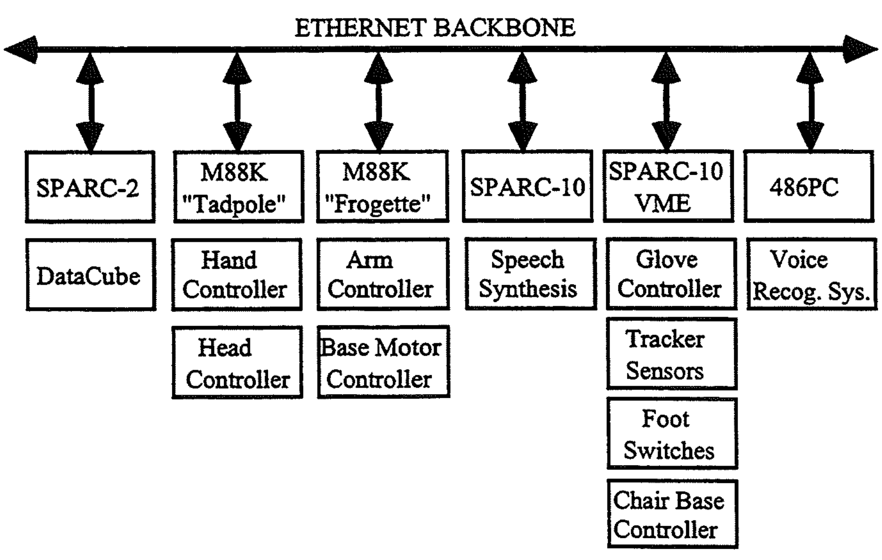


Figure 4. The DART/FITT computer configuration.

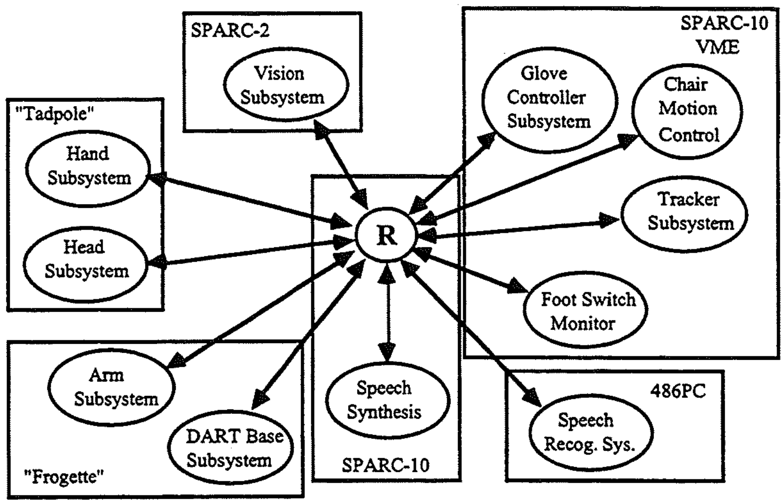


Figure 5. The distributed control architecture of DART/FITT.

OBSERVATIONS

One of the most significant observations from the preliminary evaluations is the short time it takes a new operator to become proficient with the system. For example, operators with no previous experience were able to transfer objects between the two hands and manipulate the controls on the panel within a 30 minute session. Operators with considerable experience in "cock-pit" type control have also found the training time greatly reduced due to the intuitiveness of the motion controls and the immersiveness of the visual feedback.

We have also identified several areas needing improvements. The weight of the exoskeleton glove controller caused muscle fatigue when it was necessary to maintain a specific position for a long period of time. This observation suggests the need for a mechanism that will allow the operator to re-adjust his or her arm positions (*e.g.*, indexing), and to use light-weight glove controllers.

While teleoperation of the dexterous hand offers much flexibility for grasping, it was found inadequate for manipulation. The difficulty lies in the kinematics dissimilarity between the robot's and the operator's hands.

The operator can also experience mild motion sickness when using the system due to a slight delay between the motions of the operator's head and the DART camera system. This only occurs when the operator makes large, quick head movements. Motion sickness usually occurs whenever there is a significant mismatch between the robot's and the operator's rate of motion. Motion sickness can also be caused by unintended body and head movements. However, since the operator rarely has to make large head movements once focused on a task, this problem is not a major prohibiting factor.

Although the current system provides the necessary visual cues to perform many tasks, a few limitations of the visual feedback have been observed. The visual feedback the operator receives is coarse (495 X 240 pixels) and the distance between the head cameras is a little too narrow, so the depth perception of the operator is not optimal. These visual limitations can have serious impacts on the operator's performance. For example, since FITT currently does not offer force-reflection, the operator assesses the force imparted onto the environment by watching for the amount of physical compliance. The active compliance of the DART's fingers is very useful in this regard.

Another problematic area encountered is the transformation of human hand motions to DART's hand motions. Several transformation methods were explored[6]. These methods included joint-to-joint mapping, forward and inverse kinematics transformations, and a combination of joint and Cartesian control. The two major difficulties encountered when applying these techniques are the dissimilar kinematics of the human's and DART's hands, and the slight changes in the sensor positions when the gloves are taken off and put back on. Joint-to-joint mapping was chosen as the method of control due to the computational simplicity and the intuitiveness of the control.

The telepresence evaluations also revealed some interesting operator behaviors. For example, an initial exercise is desirable before each session to familiarize the operator with the system's behavior. The exercise typically involves having the operator command the robot's arms, hands and head in various different ways to explore the dexterity of the robot. Without the exercise, less experienced operators often have the tendency to move like a robot, not fully utilizing his or her natural coordination skill. After a few training sessions, the operator generally will learn to compensate for any kinematics dissimiliarities between the operator and the robot.

Perhaps the most interesting observation was that the operator's dependency on the visual feedback decreases as a function of the amount of training time. This is most evident when the operator flipped on/off an electrical switch without actually seeing the fingertip making contact with the switch. This observation can probably be explained by the circular learning theory introduced by Piaget[5], and Held and Hein[2].

Even with the observed limitations, the original DART/FITT system was able to complete all of the assigned tasks in a reasonable amount of time.

SYSTEM MODIFICATIONS

After our initial evaluations of the DART/FITT system, we began to focus on how to overcome the limitations of telepresence without losing its benefits. As DART/FITT evolves, several features have been added to maximize the system's usefulness. These features are discussed below.

To expand the robot's capabilities beyond those of the operator's arms and hands, several features have been added. First, different voice-invoked hand grasp primitives (e.g. pinch, cylindrical, hook, etc.) were made available to the operator to compensate for the kinematic dissimilarities between the human hand and the robot hand. We have also replaced the exoskeleton glove controllers with the light weight CyberGloves™ to reduce fatigue.

Similarly, a "freeze" voice command was added to the system to enable and disable the tracking between robot and operator, allowing the operator to rest her arms. A "re-index" voice command was also added to allow the operator to control the robot in a more comfortable position. The "re-index" command also allows the operator to fully utilize the joint and reach capabilities of the robotic arms.

In a shared control scheme where the operator and the automated control primitives both have access to the robot, a method must be provided to coordinate their interactions. In the case of FITT, a speech recognition system was selected as a "hands-free" method for the operator to communicate to the robot. The operator issues commands through a microphone located on the FITT helmet. For example, the operator would say "spherical grasp" to change the configuration of the hand, or "freeze left arm" to disable tracking between the operator's left arm and the robot's left arm.

To expand viewing capabilities, a wrist camera was added to DART's right arm. The operator can switch from the "head view" to the "wrist view" for aligning the hands when grasping visually obstructed objects. Also an advanced pipeline-based vision system is being created to perform shape recognition, target location, and closed-loop visual servoing of robotic arms for grasping.

The mild motion sickness experienced by the operator when DART was rotating has been relieved by having the operator platform (a motorized chair) rotate along with the DART base. The acceleration and deceleration cues provide the operator with sufficient kinesthetic feedback to prevent disorientation.

FUTURE WORK

The early evaluations have demonstrated the versatility of the DART/FITT system and confirmed the feasibility of our approach. However, to further improve the system's productivity, the intelligent automation aspects of the system must be expanded. For example, the hand will be able to manipulate latches and handles on space hardware, as well as flexible objects such as plastic sample bags through automated sequences. Automated arm modes will be incorporated for tasks such as using a hammer to chip a rock for planetary exploration.

Several arm upgrades are planned. A control system with coordinated motion between the two arms and hands will be added to the system for use with dual arm grasps. An additional capability, scaling, where the ratio of the amount the robot moves to the amount the operator moves can be changed, will be added to the system in order to make fine motion control of the arms easier.

The advanced vision system will be enhanced to provide basic perception of the environment needed for automated manipulation and grasping. Such capability will be especially important for planetary applications since the communication delay between the operator and robot may be large.

A second generation head camera unit will be fabricated to provide tighter head tracking and to correct the narrow interpupilary distance. A high-resolution (640 X 480 pixels) head-mounted displays will be sought to improve the operator's visual acuity.

The force-reflective dexterous arm master, (Figure 2) will be integrated with FITT to evaluate the effects of force-reflection. Additional evaluations will be conducted to quantify the performance of the DART/FITT system. New test subjects will be recruited to study the correlation between training-time versus performance, and the performance of "cock-pit" type control versus telepresence.

Virtual reality simulation of the robot will be developed and over-laid into the VR helmet as a predictive display. Virtual instrument displays such as bar graphs and meters will be used to assist the operator in various tasks.

CONCLUSIONS

Telepresence is *not* a new idea. It is, however, an idea that is becoming a reality due to the recent advances in head-mounted displays, dexterous glove controllers, motion trackers, force-reflective masters, and other human compatible interactive devices. The DART and FITT combination represents an integration of these telepresence and VR technologies for space robotics applications. While further evaluations will be necessary to completely characterize the system, we believe all of our stated objectives have been met. Many lessons were learned in our preliminary evaluations and several areas for improvement were identified. Our future work will address these areas. However, the benefit of telepresence and VR in space robotics is clearly evident by the variety of complex tasks DART/FITT can perform under the control of an operator.

REFERENCES

[1] Held, R. M.; Durlach, N. I., 1992. Forum Spotlight on: The Concept of Telepresence: Telepresence. *Presence,* 1:109-112.

[2] Held, R.; Hein, A., 1963. Movement-Produced Stimulation in the Development of Visually Guided Behavior. *Journal of Comparative Physiological Psychology,* 56:872-76.

[3] Li, L., 1993. Virtual Reality and Telepresence Applications in Space Robotics. *Virtual Reality Systems Magazine,* 1:50-56.

[4] Li, L., Cox, B., Shelton, S., Diftler, M., 1994. Telepresence Control of a Dual-Arm Dexterous Robot. *Proceedings of the Third International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS 94).* Pasadena, CA. 1:313-316.

[5] Piaget, J., 1963. The Origins of Intelligence in Children. Norton, New York.

[6] Speeter, T. H., 1992. Transforming Human Hand Motion for Telemanipulation. *Presence,* 1:63-79.

[7] Wise, J. D.; Ciscon, L., 1992. *TelRIP Distributed Applications Environment Operator's Manual.* Technical Report 9103 George R. Brown School of Engineering, Rice University.

Session 3
Tracking Sensors

AN APPLICATIONS-ORIENTED APPROACH TO THE DEVELOPMENT OF VIRTUAL ENVIRONMENTS

Michael X. Crowe
GreyStone Technology, Inc.
15010 Avenue of Science, Suite 200
San Diego, CA 92128
(619) 675-7800
mxcrowe@gstone.com

ABSTRACT

The field of Virtual Reality (VR) is diverse, ranging in scope from research into fundamental enabling technologies to the building of full-scale entertainment facilities. Due to the multi-faceted nature of this field and complicated by excessive media attention and interpretation, the concept of virtual reality means many things to many people. Ideally, a definition of VR should derive from how this technology can provide solutions to existing challenges in building advanced human-computer interfaces. The measure of success for this technology lies in its ability to enhance the assimilation of complex information, whether to aid in difficult decision-making processes, or to recreate real experiences in a compelling way. This philosophy and the virtual environment development process employed by the engineers and artists at GreyStone Technology, Inc. is described using an example from a VR-based advertising project. The common and unique elements of this example are explained, though the fundamental development process is the same for all virtual environments that support information transfer. In short, this development approach is an applications-oriented approach, one that begins by establishing and prioritizing user requirements and seeks to add value to the information transfer process through the appropriate use of VR technology.

INTRODUCTION

This paper describes the development process used by GreyStone in the creation of virtual environments to support complex information transfer. Rather than focus on iterative improvements in display technology or image generators, what is presented here is a discussion of how the initial design of a virtual environment must be geared to the ultimate application of the system and how the various technology components that underlie that system are integrated to provide a working, value-added interface. The emphasis here is on recognizing the unique value of this medium for information transfer and maintaining product focus throughout the development process. This "applications-oriented" process is not specific to certain applications, but is adapted and modified depending on the requirements of the end-user and is driven by factors such as current and emerging capability, affordability, and the accomplishment of specific application objectives. The kinds of applications that have been developed using this process include planning, training, and entertainment experiences.

The Value of Virtual Reality

First, it is useful to put this discussion in the proper context and provide some background. GreyStone is in the business of creating information processing and information transfer products. The company recognizes that information is a critical commodity - the commodity of the future. This focus on information is shaping the nature of business across the globe, and the possibilities for both providers and consumers is enormous. Today, the sheer volume of information available to consumers and businesses is staggering, and this level continues to increase exponentially. But, having access to a large volume of data is of no value unless the information is presented in a usable form. That is where virtual reality comes in: VR technology allows us to build systems that put information into a usable form and make it accessible to the user in an effective way.

It is hard to underestimate the importance of information to human society. In fact, it could be argued that the ability to record and transfer information is the primary ability that distinguishes humans from all other animals [1]. Being able to represent ideas and observations as recorded symbols gave our early ancestors a major competitive advantage. And though other animals can use tools or communicate via language, only our species is able to capture important information in symbolic form to be passed on to future generations. So, the current information revolution is not the first, but one in a series of societal shifts that have influenced our evolution. The current "information age" is about the ability to put information into digital form, to process that data very quickly via computer, and to distribute information at the speed of light to everybody on the planet (and beyond).

Information, then, is a *representation* of something else. In its representational form, it is inert, dependent on somebody or something to *interpret* its meaning and complete the transfer process. With the recent ability to rapidly create and copy information of high volume and complexity came the requirement to complete the information transfer process in more effective ways. The proper employment of VR technology allows this to be accomplished by presenting information at a lower level of abstraction than typically used by traditional computer interfaces. This means that a VR interface presents information in a form that is directly interpreted by the human senses, like images, sounds, and motion. Furthermore, these interfaces allow visible and invisible phenomena to be intermixed, correlated, and displayed. Such an interface may one day allow air traffic controllers, surgeons, and mission planners to better understand their complex, multi-dimensional problem spaces and allow them to make better decisions in less time. If the data representation and transfer process in a VR interface is accurate enough, the participant may get the sense of actually "being there". This fact has led a number of developers to extend the application of VR to the entertainment and advertising markets.

THE APPLICATIONS-ORIENTED DEVELOPMENT PROCESS

Creating an effective virtual environment requires the coordinated participation of a multi-disciplinary team and the orchestration of that team throughout the many stages of the development process. To ensure that the resulting system achieves its expected aims, it is crucial that an applications-oriented approach be used during the project definition phase, and that this definition then serves as a functional guide throughout the implementation process. A specific VR development project is used here as an example to illustrate points from the process description. Though any number of examples could have been chosen, this one demonstrates some unique requirements that had to be considered in the design phase. The example is an experience called *Virtual Voyage*™, which was created for product promotion purposes. In this experience, participants navigate a clipper ship from one port to another to deliver a cargo of whisky.

Step 1 - Understand the User's Requirements

Perhaps the most critical step in designing an effective virtual environment is to put oneself at the end of the process and analyze how the system will be employed by the ultimate user. What are the crucial elements of information that must be present? What are the functions that the interface must provide? What is the background of the expected operators? Clear answers to these questions may indicate that a fully immersive environment is not desirable at all, which would have a drastic impact on the system design process. Other user-dependent questions will help frame the development process, like: How important is domain expertise in the creation of an effective interface? What is difficult about accomplishing this interface task using traditional techniques? What is the available budget? What kinds of ergonomic issues exist? By clearly addressing these sorts of questions during the project definition phase, the developed system takes on an applications-oriented purpose, helping to guide its evolution and ensure its usefulness.

In the case of *Virtual Voyage*, the design of the system was driven not only by the general public that would eventually experience the system, but primarily by the customer: the advertising agency paying for the development of the system. Obviously, the advertiser's goal is to feature the product in a relevant and entertaining way. Since the product being advertised is an alcoholic beverage, it was decided to create an experience from the days of prohibition, a recreation of the ferrying of liquor from the Bahamas to Long Island aboard the clipper ship of Captain McCoy. This *theming* of the experience is extremely important for giving the participants a context for involvement, and in this case, helping to achieve the advertiser's goals. At this point, it becomes evident that a certain amount of domain experience will be required to ensure realism of the sailing vessel's behavior and responses. With the purpose of the virtual environment clearly in mind, the entire experience can now be created around the established theme.

Another important factor at this stage is a thorough understanding of the eventual participants who will experience the virtual environment. The users of the system are members of the general public, who happen to see this advertising event as it tours the country. It was desired to leverage the sense of immersion and interactivity with the environment that VR gives the participant, while maintaining the focus on the product being advertised. Since the users would likely experiencing VR for the first time, it was also important to keep the game concept straightforward and the interface simple. Further, the system must achieve its objective within just a few minutes of play. To achieve these aims and promote product identification, the general concept developed was for the participant to sail the virtual vessel while protecting the cargo from various hazards along the way. These hazards could be in the air, on the water, or on the ship itself, so the visual interface would have to provide the participant with a full field of regard. A hand-held gun with a virtual representation would allow the participant to defend his or her cargo, thereby linking the participant with the scenario and building an identification with the product.

Step 2 - Identify Where and How VR Adds Value

When designing an interface for information transfer, one must be careful that the technologies chosen to support the interface actually enhance the transfer process rather than distract from it. The goal should be to effectively transfer the desired information, not showcase the latest VR product or technique. A big part of virtual environment development has to do with understanding how human beings detect and assimilate sensory information and what tools and techniques are available to reproduce these effects [2]. Of course, a certain amount of cost realism usually enters the equation, but even with compromises, a compelling, immersive environment can be achieved. The key requirements are to provide a high-fidelity visual scene, to correlate that visual scene with an appropriate soundscape, and to provide at least some level of somatic contact with the system (anything from a simple joystick interface to a motion-base), which also allows the participant to interact with the virtual environment. Additionally, this must all be accomplished in a small enough time frame that the user does not notice things like system latency or update rates. Though other sensory modalities can be brought into play, the cost is likely to outweigh the benefits. Again, the analysis should be based upon the requirements defined in step one.

Having settled on the deck of an open sailing vessel as the environment to be recreated in *Virtual Voyage*, the primary requirement is to create a large and realistic visual scene. The fact that events in the environment are occurring all around the participant makes it a good candidate for a head-mounted display (HMD) with a head tracking system. Additionally, the visual resolution requirements in this case are lenient enough to allow an HMD to be used cost-effectively. HMDs are not always the best display medium, and their use certainly does not define the interface system as a virtual environment. Going back to the original guidelines for design, it was determined that a repeat monitor must be included to allow a larger audience to share what the participant sees within the HMD. This allows the information transfer process to reach a much larger audience.

Somatic contact would be achieved through a physical mockup of the vessel's helm and a plastic pistol. In fact, a portion of the deck and mast was eventually built to provide continuity of the theme between the real and virtual worlds. The design allowed the participant to actually steer the vessel by turning the wheel and observing the change in attitude relative to the prevailing wind in the sails and the motion of the waves. By way of a motion-tracker attached to the hand-held pistol, a virtual gun could be moved about the virtual space. These two modes of physical interaction were intended to get the participant's body involved in the experience, helping to close the sense of presence.

Finally, the auditory channel would have to be supported via a detailed sound profile of the dynamic environment. The interfaces chosen were headphones to transmit the soundtrack to the participant and loudspeakers to repeat those sound to the audience. To enhance the sense of immersion, a three-dimensional sound system was selected to allow the participant to localize sounds in the surrounding virtual space. Other sensory pathways could have been engaged, like wind blowing from the appropriate direction or a motion-base to simulate the toss of the ocean, but these were not included since the desired effect was achievable at a reasonable cost using the above components.

Step 3 - Develop Concepts of Operation

The last step before implementation can begin is to fully scope the operational employment of the virtual environment. Based on the preliminary definition established thus far, and keeping the user's requirements in mind, the "flow" of system operation must be defined. For entertainment-oriented experiences, this would involve the development of an appropriate storyline, perhaps supported by artistic renderings of probable scenes and scenarios. More serious applications, like a mission planning system, would use sample scenarios to exercise the sequence of interactions during the planning process. Building on the basic goals and components established in steps one and two, the theme must now be "filled out" with details and complete concepts.

In our example process, the operational sequence of *Virtual Voyage* was completed through the development of a few scenarios. An "objective" for the experience was defined: the participant would be scored based on the number of cases of scotch that were successfully delivered to the final port. The primary hazard to delivery would be a stowaway who would attempt to steal cases from the stack of cargo. This virtual thief could be (non-lethally) shot to discourage his troublesome activity. The stowaway would appear throughout each of the voyage scenarios. The scenarios involved navigating the vessel out of the natural harbor in the Bahamas (the initial scenario), and engaging opponents on the open ocean. During a storm scene, good sailing skills would reduce the loss of cargo over the side of the ship. On the way to "Gatsby's Mansion" on Long Island, the participant would encounter a rival gang in speedboats who shoot at the cargo to reduce the player's profits and an attack seaplane with a similar objective. By shooting these opponents, the participant ensures the safety of his or her cargo and thereby achieves a higher score.

The choice of interface devices in step two has implications in the development of an operational concept. A detailed visual representation of the surrounding world meant that a substantial amount of physics would have to underlie the behavior of the graphical objects. This not only applies to the vessel being sailed by the participant, but to the computer-generated adversaries. Of course, attention would have to be given to the polygonal representation of the visual scene, the use of appropriate image textures, and the scene's depth complexity. Additionally, each scenario would have an associate soundtrack so that objects and interactions were represented aurally. This, too, would be of high fidelity, with sound localization included. The kinds of sounds to be sampled and created (ocean waves, seagulls, creaking of the masts and rigging, gunshots, etc.) were identified at this point.

Each of the scenarios or vignettes was defined in this way to serve as a development template during the implementation phase. The final scenario was defined as the arrival at "Gatsby's Mansion" (to some considerable fanfare, depending on the amount of cargo delivered). Following the initial departure scenario, the vignettes are chosen randomly to provide a sense of novelty for viewers who see the game played a few times. A consistent storyline was put together, and the timing of vignettes and the overall experience were established (30-45 seconds for each vignette and approximately 3 minutes overall).

Step 4 - Implementation and Integration

Finally, the system must be constructed based on the foregoing design. Clearly, this is the bulk of the development process, though, unfortunately beyond the scope of this paper. If the project definition phase has been accomplished successfully, it will serve as a constant guide throughout the implementation process. Having established an applications-oriented goal for the virtual environment system, the following steps can be carried out to achieve project completion:

- Artistic Rendering of Objects and Scenes
- Simulation of Underlying Physics and Object Relationships
- Modeling of Entity Behaviors and Interactions
- Integration of the Human Participant with the Virtual Environment

A photograph of a participant engaged in the completed *Virtual Voyage* experience is shown in Figure 1 and an image of the virtual environment itself is shown in Figure 2.

CONCLUSION

The development of a virtual environment is a complex undertaking requiring skills from many creative and engineering disciplines. To orchestrate these skills and keep the development process focused on the ultimate purpose of the information transfer system, it is important to adopt an applications-oriented methodology. This process consists of understanding the user's breadth of requirements, recognizing the value of VR technology for the current application, applying the technology appropriately, and creating a detailed concept of operation prior to implementation. If such a methodology is followed during the design phase, the implementation task is more focused and directed, and the resulting system is more likely to achieve its objectives.

REFERENCES

1. Rothschild, Michael, "Cro-Magnons' Secret Weapon", Forbes ASAP, New York, September 13, 1993, pp. 19-20.

2. Robinett, Warren, "Synthetic Experience: A Proposed Taxonomy", Presence, Massachusetts, Volume 1, Number 2, Spring 1992, pp. 229-247.

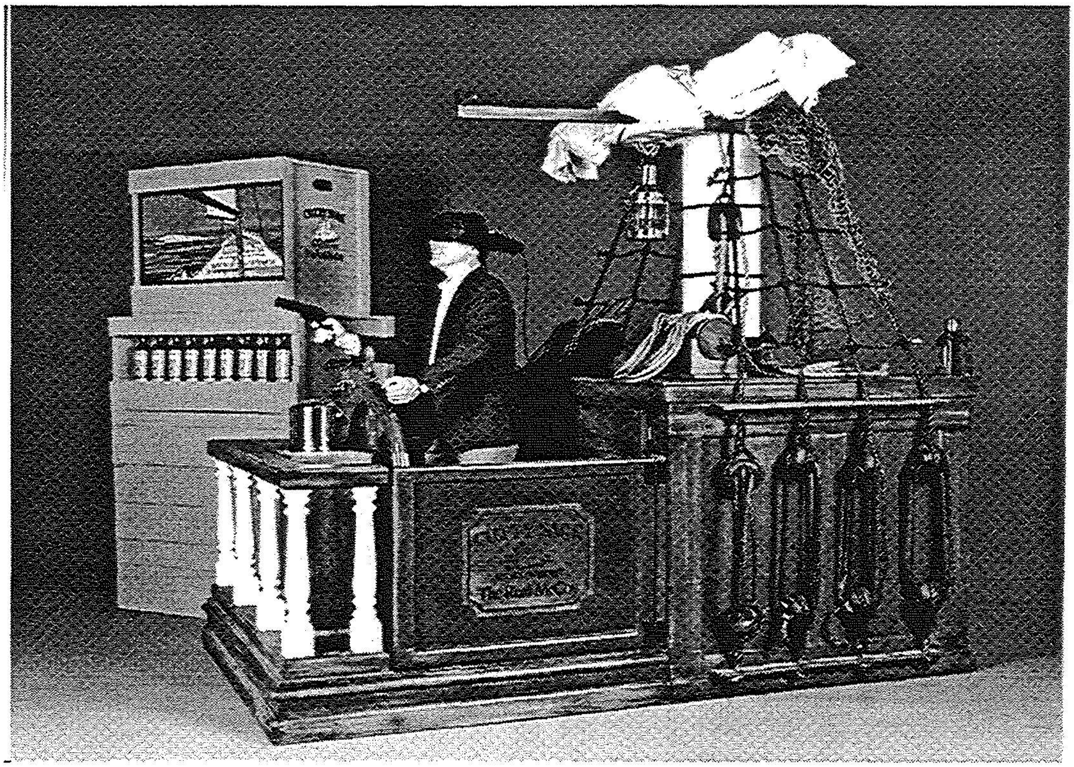


Figure 1. A participant immersed in the *Virtual Voyage* experience.

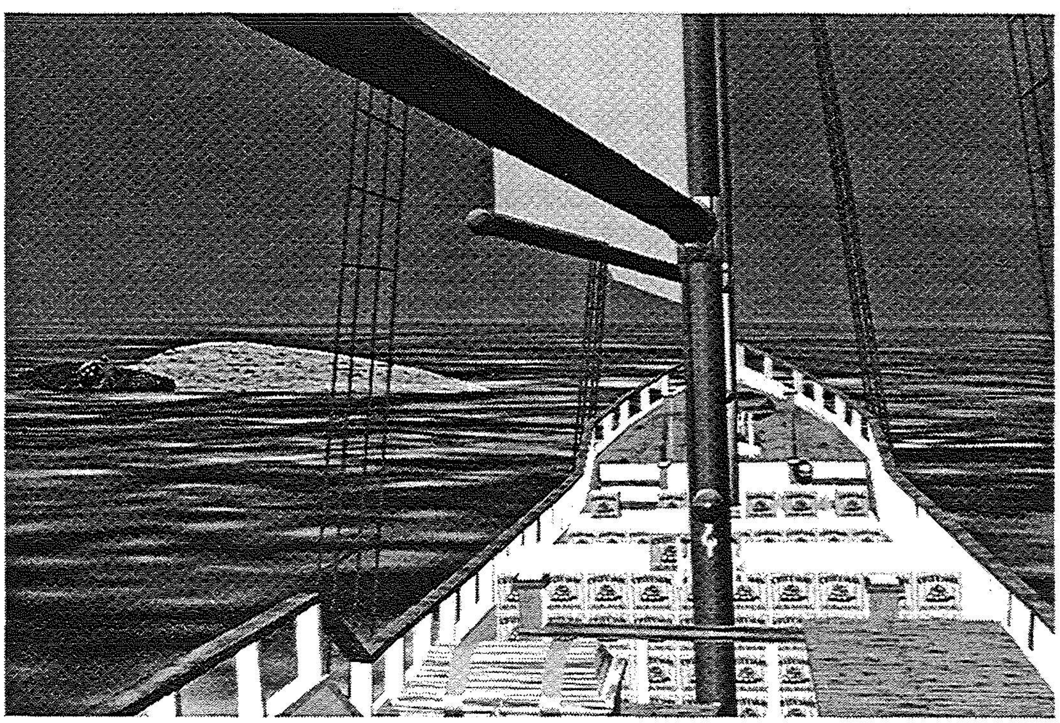


Figure 2. The stowaway and speedboats harass the participant in *Virtual Voyage*.

N95- 15981

A SPECIFICATION OF 3D MANIPULATION IN VIRTUAL ENVIRONMENTS

S. Augustine Su and Richard Furuta
Hypermedia Research Lab
Department of Computer Science
Texas A&M University
College Station, Texas 77843-3112
E-mail: {su, furuta}@bush.cs.tamu.edu

ABSTRACT

Virtual Reality techniques have promised intuitive and effective user interfaces to virtual worlds. The use of hand gestures is an important part of that interface. However, due to the absence of maturity of standard and tailorable software abstractions such as those seen in 2-D graphical user interfaces, current techniques for specifying the interactions of 3-D objects and gestures are ad hoc and indirect.

In this paper, we discuss the modeling of three basic kinds of 3-D manipulations in the context of a logical hand device and our Virtual Panel Architecture. The logical hand device is a useful software abstraction representing hands in virtual environments. The Virtual Panel Architecture is the 3-D counterpart of the 2-D window systems. Both of the abstractions are intended to form the foundation for adaptable 3-D manipulation.

Within our software framework, the click-and-drag operation from the 2-D graphical user interface context gracefully can be replaced by a meaningful hold-and-move operation for applications in virtual environments. With these tailorable abstraction tools, the semantics of natural and precise gestures can be prototyped rapidly.

INTRODUCTION

Incorporating gestural control into Virtual Reality environments holds the promise of providing intuitive and effective user interfaces to interact with virtual worlds. By using their hands to directly manipulate 3-D objects, the environment's users have the potential to gain much more freedom than in the traditional 2-D mouse and keyboard environments. However, due to the absence of maturity of standard and tailorable software abstractions, current 3-D manipulation techniques are ad hoc and indirect when compared to 2-D graphical user interfaces. Furthermore, since 3-D manipulation is still far from fully explored, the complexity with which current environments permit interactions between the user's hands and 3-D objects is still very limited.

There are two major paradigms for the use of hands in virtual environments. The first paradigm is to point, shoot, or grab 3-D objects. This manipulation method is directly generalized from the use of a 2-D pointer, and can be implemented by a 3-D mouse with buttons, which has the ability to detect positions and orientations in 3-D space. These gestures can be combined with other sources of input; for example, human speech can be combined with gestures to specify quantities as in [1, 2]. In this situation the gestures act as 3-D pointers, and the speech acts as buttons to signify status changes when the hands are not available to push buttons. It is clear that this first paradigm is very useful, but, however, does not take full advantage of the freedom given it in 3-D space.

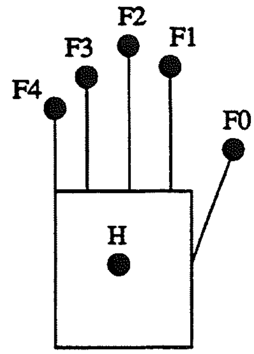


Figure 1: The six points of interest on a hand for the hand device

The second paradigm is to create sets of static or dynamic gesture commands for specific applications as in [3, 4, 5]; each gesture represents a single command with pre-defined semantics in the context of applications. The gestures in this paradigm do not necessarily correspond to physical manipulations—indeed as one example, interfaces can use gestures borrowed from a sign language such as American Sign Language.

Ideal 3-D user interface models have to be able to accommodate not only the above approaches, but also to provide tailorable tools for new user interfaces to meet various needs. We believe we have found a good user interface model for 3-D manipulation. In this paper, we will discuss the modeling of three popular gestures based on a logical hand device and the Virtual Panel Architecture of our work. With proper abstraction tools, the semantics of natural and precise gestures can be prototyped rapidly.

In the next two sections the hand model and the Virtual Panel Architecture will be briefly discussed, respectively. Afterwards, three popular gestures will be described based on the hand model and the architecture.

THE LOGICAL HAND DEVICE

The innovation of logical devices in a graphics package is to conceal discrepancies among disparate physical devices of a kind, and to furnish device-independent characteristics to application programmers.

By the same token, the logical hand device [6] was designed to be a useful software abstraction representing hands in virtual environments. The hand device reports hand information in the form of events to the system. The hand information consists of

1. the positions and orientations of the five digit tips and the center of the back of the hand (Figure 1); that is, the output of six 3-D mice, or six 3-D pointers.
2. digit-oriented handshape features, such as straight, flat, curved, fully curved, and so on for each finger, and adduction or abduction for adjacent fingers. These features can be used to compose American-Sign-Language-like static gestures.

With this hand device, we can meet the need of the two major paradigms of using 3D gestures in virtual environments: the style of "point, reach, and grab" and the command by sign-language-like gestures.

THE VIRTUAL PANEL ARCHITECTURE

The principle of the manipulation in 2-D graphical user interface is to use a single 2-D pointer to move into and out of a number of hierarchical 2-D windows, and to use mouse buttons to signify status changes. Based on that, higher-level tasks, such as click-and-drag, can be implemented. This 2-D manipulation methodology can be generalized for 3-D manipulation. Think about the use of hands or fingertips to directly manipulate 3-D objects while the hands are characterized by the logical hand device. The hand device provides the concept of multiple pointers and gesture features. These pointers are directly mapped to the points of interest of the manipulation. Those composable gestures can form a base to signify various status changes.

With the above philosophy in mind, a software framework—the Virtual Panel Architecture [7]—was designed to help implement an intermediate abstraction for the manipulations of 3D objects by hand gestural input. There are three major components in the architecture (see Figure 2): the *Gesture Server* is responsible for extracting information from physical hand tracking devices and composing gestures for the use of a later stage; the *Panel Server* is in charge of maintaining a database of 3-D objects, and of reporting interactions by multiple pointers in the form of events; and the *filtering processing stage* is used to encapsulate information from the events to be sent to application programs.

SPECIFICATION OF GESTURES

In this section three basic gestures, touching, pointing, and gripping, will be discussed in the framework of the hand device and the Virtual Panel Architecture.

A gesture can be as simple as *touching*: no extra specification is needed. A gesture can be fully specified in the Gesture Server as *pointing*: here digit-oriented handshape features play the major role to define the gesture. Or, a gesture can be fully specified in the Panel Server as *gripping*: in this case the interactions of objects and pointers are concerned. These three gestures demonstrate the usability and flexibility of our framework.

Touching

The simplest gesture is touching, that is, a 3-D pointer enters the territory of an object. It is the Panel Server's responsibility to detect the invasion of a pointer into an object, and then to report events to a filter associated with the object.

Pointing

Pointing is a gesture with a specific handshape. One of the possible ways to define pointing is as below: (1) fingers except the index one are "fully curved" and are "enclosed" by the thumb; (2) the index finger is "straight" or near straight; (3) probably, we want to restrict the orientation of pointing gesture within some range (the terms enclosed by double quotes are features in the digit-oriented handshape alphabet.). The gesture is detected by the Gesture Server if we have registered the gesture in the Server beforehand. As a result the position of the index fingertip is the starting point of the pointing; the orientation of the index fingertip is the pointing direction. Both of the values are sent to the Panel Server, which has to detect the shooting target from the fingertip information.

Gripping

Another important gesture is gripping gesture. With this gesture the click-and-drag in 2-D graphical user interface can be superseded by the hold(grip)-and-move in 3-D space.

In the beginning the concept of click is replaced by that of *holding*. A 3D holdable object has to be specified by a set of points, edges or faces which are *holdable* places on the object. When one or more fingertips and the thumb tip enter the holdable places of an object, then we regard

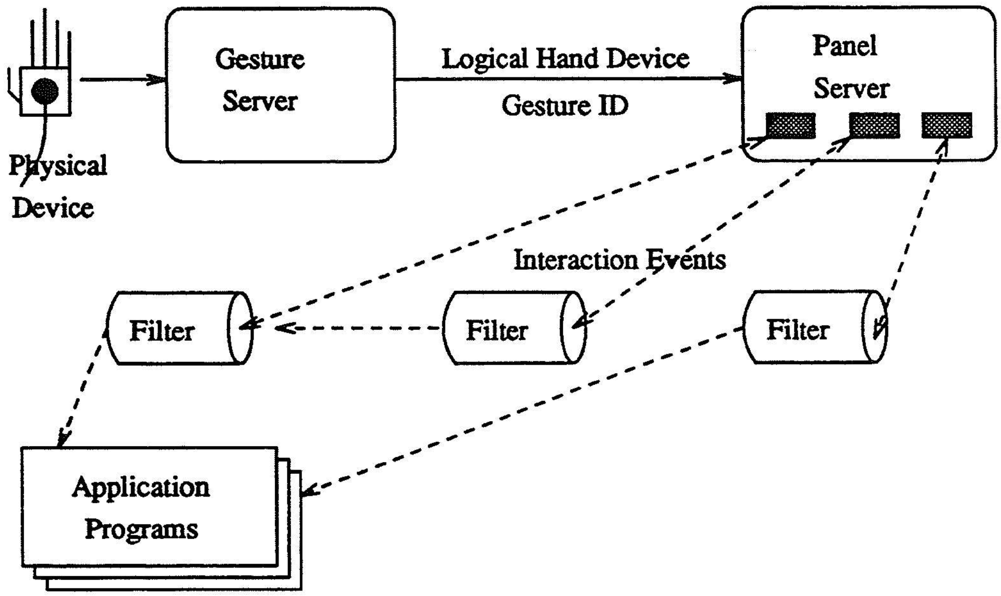


Figure 2: The Virtual Panel Architecture

the object as *being held*. The whole holding process is handled by the Panel Server, which knows that pointers are entering holdable objects. We also can release an object by letting less than two pointers stay in the holdable places of the object. As long as pointers are holding an holdable and movable object, the object can be moved around in 3D space by the hold-and-move.

An object can define its own action rules in its associated filter to react to various holdings. The *holding* can mutate with *Tip Grip*, *Pinch Grip*, *Lateral Pinch* [8], etc. to signify different states as different mouse-button combinations.

CONCLUSION

The advantages of the above user interface model in virtual environments are three-fold: the user can concentrate on limited parts of interest on the hands while the major semantics of gestural interactions are still maintained; application programmers can focus on these salient points only to simplify programming jobs; and, the computation load in the system will be relieved since the detection of precise contacts of hands upon 3D objects will be reduced from computing a whole hand into computing a number of points only.

Currently we are experimenting with the framework using a VPL DataGlove, which is connected to a Macintosh and a SPARCstation. The DataGlove does not have the power to extract all of the information on the logical hand device. However, the partial information on the hand from the DataGlove gives us a good beginning.

Our modeling of the gestures has shown that the expressive power of our user interface model is at least not less than that of a 2-D graphical user interface because of the hold-and-move operation. However, there is still a broad space in 3-D manipulation that has not been explored, especially for multi-pointer interactions. We continue the study on the model to determine if it is able to accommodate new and novel interactions. We hope this line of research will eventually benefit the standardization of 3-D manipulation in virtual environments.

ACKNOWLEDGEMENT

This material is based on work supported by the National Science Foundation under Grant No. IRI–9496187.

REFERENCES

1. Hauptmann, A. G., and McAvinney, P., "Gestures with Speech for Graphic Manipulation", Int. J. MAN-MACHINE STUDIES, Academic Press, Vol. 38, 1993, pp. 231-249.

2. Bolt, R. A., and Herranz, E., "Two-Handed Gesture in Multi-Modal Natural Dialog", Proceedings of the ACM Symposium on User Interface Software and Technology, Monterey, California, November, 1992, pp. 7-14.

3. Baudel, T., and Beaudouin-Lafon, M., "Charade: Remote Control of Objects Using Freehand Gestures", COMMUNICATIONS of the ACM, Vol. 36, No. 7, July, 1993, pp. 28-35.

4. Figueiredo, M., Bohm, K., and Teixeira, J., "Advanced Interaction Techniques in Virtual Environments", COMPUTERS & GRAPHICS, Pergamon Press, Vol. 17, No. 6, 1993, pp. 655-661.

5. Papper, M. J. and Gigante, M. A., "Using Gestures to Control a Virtual Arm", VIRTUAL REALITY SYSTEM, Academic Press, 1993, pp. 237-246.

6. Su, S. A., and Furuta, R., "A Logical Hand Device in Virtual Environments", in VIRTUAL REALITY SOFTWARE & TECHNOLOGY: proceedings of the VRST'94 conference, Singapore, August, 1994, pp. 33-42.

7. Su, S. A., and Furuta, R., "The Virtual Panel Architecture: A 3D Gesture Framework", in the proceedings of the 1993 IEEE Virtual Reality Annual International Symposium (VRAIS), Seattle, Washington, September, 1993, pp. 387-393.

8. Kroemer, K. H. E., Kroemer, H. J., and Kroemer-Elbert, K. E., ENGINEERING PHYSIOLOGY: Bases of Human Factors/Ergonomics, 2nd ed., van Nostrand Reinhold, New York, 1990, p. 22.

Session 4
Displays

TELE HYPER VIRTUALITY

Nobuyoshi Terashima
ATR Communication Systems Research Laboratories
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02, JAPAN

Abstract— In the future, remote images sent over communication lines will be reproduced in virtual reality(VR). This form of virtual telecommunications, which will allow observers to engage in an activity as though it were real, is the focus of considerable attention. The system will offer the experience of being in a place without having to physically go there.

Taken a step further, real and unreal objects will be placed in a single space to create an extremely realistic environment. Here, imaginary and other life forms as well as people and animals in remote locations will gather via telecommunication lines that create a common environment where life forms can work and interact together. Words, gestures, diagrams and other forms of communication will be used freely in performing work.

Actual construction of a system based on this new concept will not only provide people with experiences that would have been impossible in the past, but will also inspire new applications in which people will function in environments where it would have been difficult if not impossible for them to function until now.

This paper describes Tele Hyper Virtuality concept, its definition, applications, the key technologies to accomplish it and future prospects.

INTRODUCTION

In the future, remote images sent over information super highways will be reproduced in virtual reality(VR). This form of virtual telecommunications, which will allow observers to engage in an activity as though it were real, is the focus of considerable attention. The system will offer the experience of being in a place without having to physically go there.

Taken a step further, real and unreal objects will be placed in a single space to create an extremely realistic environment called Hyper World. Here, imaginary and other life forms as well as people and animals in remote locations will gather via super highways, to a common environment called The Coaction Environment, where life forms can work and interact together. Words, gestures, diagrams and other forms of communication will be used freely in performing work.

Actual construction of a system based on this new concept will not only provide people with experiences that would have been impossible in the past, but will also inspire new applications in which people will function in environments where it would have been difficult if not impossible for them to function until now.

This paper describes the concept, the technologies accomplishing it and the future prospects.

CONCEPT OF TELE HYPERVIRTUALITY

Inhabitants, such as people and animals in remote locations as well as imaginary and other life forms, will be able to coact; that is, they will be able to work and interact together, in a Hyper World where real, unreal and other worlds are fully integrated.

Hyper World

Hyper World is an advanced form of reality where real-world, computer graphic and other images are systematically integrated. Here, real-world images shot by camera and recognized by Computer Vision(CV) are realistically reproduced in Virtual reality(VR). These images may then be sent from remote location via super highways.

Coaction environments

Inhabitants, such as people, animals, and imaginary and other life forms, will be able to work and interact together using words, gestures and other forms of communication in the Hyper World environment. This interaction is referred to as coaction.

Coaction not only allows people in remote locations to work and play together as though they were in the same room, but it also allows people to interact with imaginary life forms.

(1) Definition of a coaction environment

This highly realistic environment provides interrelated objects with a common site, that is, a workplace or an activity area. The environment offers a means through which activities such as designing buildings, sharing activities and playing catch can be performed while communicating through words and gestures. The manipulation of physical bodies not only requires that objects conform to the laws of physics, such as moving like they are supposed to move, and changing shape when they collide, but that various life form activities take place, such as plants wilting and blooming according to their exposure to sunlight.

New environments are created by the interaction of coaction environments. In other words, multiple independent coaction environments interacting to form an integrated coaction environment exchange knowledge for integration. Sometimes these environments return to their original state after they stop interacting.
Dynamic changes such as those just described are a feature of coaction environments.

(2) Advanced operation

The basic function of an advanced operation is to enable activities such as automobile design or a recreational event using words, gestures, diagrams, voice and other forms of communication.

(3) Coaction control

This provides control functions and a site for interactive work and activities. The control functions include common area control, as well as integration, separation and other common area activities corresponding to the interaction of the inhabitants.

Definition

Technologies for creating highly realistic environments

S: Highly realistic environment

This is defined as follows.

$S = \{S_E, S_{CA}, S_{CG}, S_{CV}\}$

S_{CA}: Nature, buildings and other objects shot with a camera

S_{CG}: Objects created through computer graphics

S_{CV}: Nature and other objects recognized and reproduced by Computer Vision

S_E: Real objects

Definition of inhabitants

I: People, animals, and imaginary and other life forms found in S

This is defined as follows.

$I = \{I_E, I_{CA}, I_{CG}, I_{CV}\}$

I_E: Real people and animals

I_{CA}: People and animals shot with a camera

I_{CG}: Imaginary life forms created using computer graphics

I_{CV}: People or animals recognized and reproduced by Computer Vision

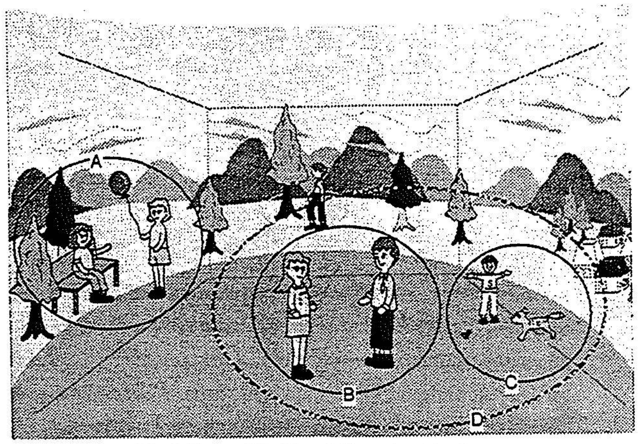


Fig.1 Hyper World and Coaction Environment

Coaction environments(CE)

A coaction environment(CE) is a group of individual coaction environments(CEi) in a realistic space(S) where inhabitants(I) are coacting.

CE = {CEi}

A CEi is an individual environment where inhabitants(I) are coacting.

Individual CEi dynamically integrate and separate repeatedly through CEi interaction. Coaction environments are shown in Fig. 1.

A, B and C are coaction environments.

If we assume that the ball in C rolls into B, the human figure in B picks up the ball and returns it to the human figure in C, then environments B and C are integrated to form coaction environment D. When the human figure in C accepts the ball and starts playing with the puppy, environments B and C are separated once again.

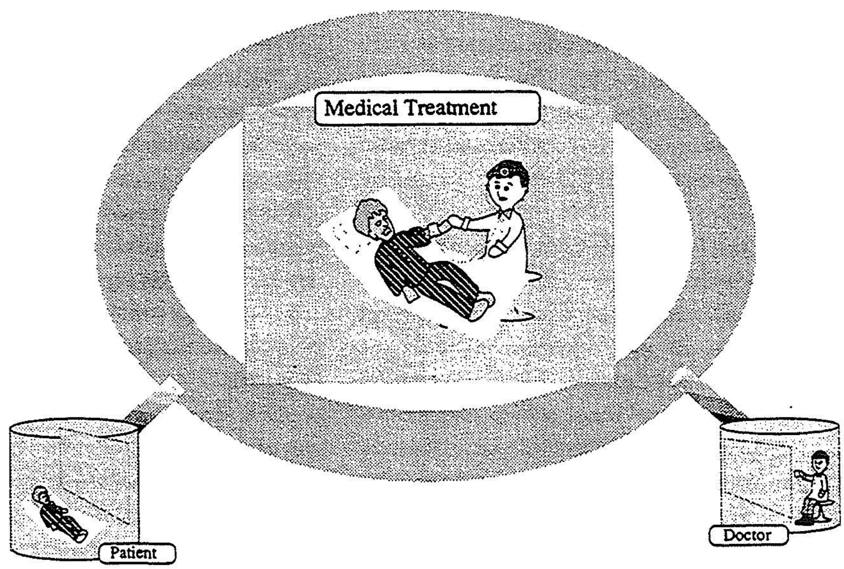


Fig.2 Medical Care

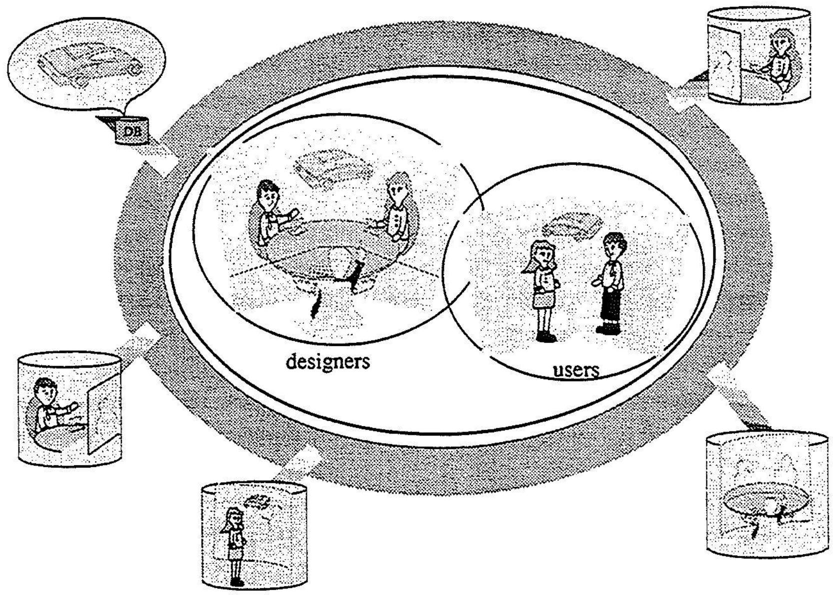


Fig.3 Coaction Environment

Applications

Tele Hyper Reality enables coaction activities in highly realistic environments connected by communication lines.

This open up possibilities for a variety of applications, ranging from medical applications, such as home care and home medical treatment for aging societies; various types of design work applications, such as automobile design; educational applications, such as remote classes and remote experiments; and entertainment applications, such as games and recreation. An image of medical treatment is shown in Fig. 2. An automobile design coaction is shown in Fig. 3.

Implementation

Real-time object image recognition

(1) Image recognition

In order to display images of a highly realistic environment from the observer's perspective, cameras are placed around the targeted natural and physical objects, and a method for switching the cameras position according to observer's perspective is being considered. In actuality, this method does not offer sufficient realism because of the lack of continuity during image switching.

To overcome these problems, the images of targeted natural and physical objects are first placed into a computer using Computer Vision, and then a method that creates from the observer's perspective and displays images in real time is used.

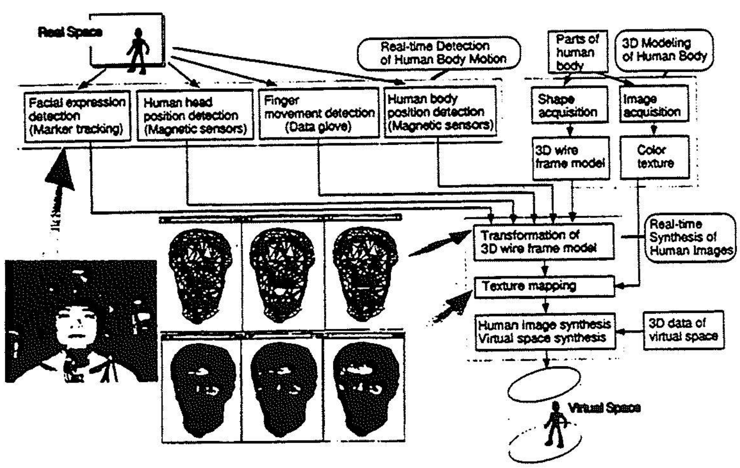


Fig.4 Real time recognition and generation of human image

This requires the technology to recognize targeted images, together with image creation and display technologies that create and display appropriate images, viewpoint and perspective detection technologies that detect the observer's perspective, and other technologies.

Fig. 4 provides details about the above technologies as they relate to human figures. Human figures are recognized and the wire frame model and texture are obtained and stared in the work station. Human figures are easy to model because they have many features in common, but natural and other physical objects are quite difficult to model because they come in all shapes and sizes. If a model can be created, the model is treated like a mannequin, and is then an easily recognizable target. However, new technologies will have to be developed in order to recognize natural and other physical objects that are difficult to model.

(2) Recognizing movement information

Information for head, hand and other movements by human figures will be detected in real time. Research is currently underway on non-contact movement detection methods as well as on detection methods in which sensors are placed on numerous parts of the human body.

Movement information for natural and other physical objects, on the other hand, is not that easy to detect. If we target a single tree, for example, some method will have to be found to detect the movement of each individual branch.

(3) Creating images

Information related to targets acquired in (1) as well as information related to movements acquired in (2) are used to create targeted images in real time using computer graphics.

(4) Displaying images

Images created in (3) are displayed on a large screen.

Since shutter glasses are used to create a three-dimensional perspective, images corresponding to the left and right lenses are switched and displayed at high speed. A three-dimensional image is obtained by viewing the image with shutter glasses.

A three-dimensional image will be achieved with the naked eye using a lenticular screen method that displays images corresponding to the left and right eye viewpoints, respectively, through 3.6-mm slits. Lenticular screen method is shown in Fig. 5.

Coaction environments

The next step will be to create a coaction environment where inhabitants will work together and take part in recreational activities, such as playing catch. In the coaction environment, object manipulation by words and gestures is used to play catch, to perform design work, and to control physical movement according to the laws of physics. If coaction environments interact, an entirely new coaction environment is created in line with the specific form of interaction.

(1) Changes in the coaction environment due to interaction

Once two environments A and B interact, a new environment that includes both the original environments is created.

Coaction then occurs in the new environment. Fig.1 shows the merging of two environments.

(2) Basic manipulation, laws of physics, biological laws

An object manipulation method based on words, gestures and other forms of communication will be used. Here the laws of physics, illustrated by objects falling, dishes breaking, physical changes from objects colliding, as well as sounds, are faithfully reproduced. Biological laws, such as plant growth and wilting in sunlight, are also faithfully reproduced.

Future Prospects

We have proposed a coaction environment in which people, animals, imaginary and other life forms work and play together in a highly realistic environment that includes real as well as unreal objects. These objects may be sent over communication lines from remote locations.

By providing an environment that goes beyond reality as just described, people will be able to perform work and activities not possible in a real environment. This system will contribute tremendously to the welfare of humankind.

An example might be providing all-night care for invalid elderly family members by artificial life forms that will notify the family, and

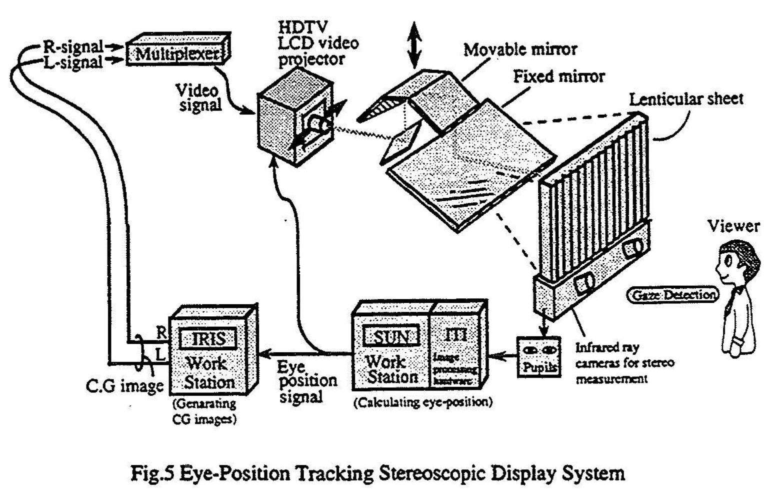


Fig.5 Eye-Position Tracking Stereoscopic Display System

take appropriate action at the appropriate time when there is a problem. Such a system will also provide heretofore unfathomable benefits, like helping people to function in particularly difficult environments, such as underwater, underground and in outer space.

References

[1] N. Terashima,"Basic Research on an Intelligent Communication System," IEEE Denshi Tokyo, 1992, pp.18-22, No.31.

[2] N. Terashima,"Cooperative Work in Teleconferencing System with Realistic Sensation," Hypermedia '92 Bepu Bay Conference report,1992, pp.115-119.

[3] H. Takemura,"Cooperative Work Environment Using Virtual Workspace," Proc. of Virtual Computing,1992,pp.171-181.

[4] S. A.BENTON et. al.,"Interactive Computation of Display Holoranis," Proc. of Virtual Computing,1992, pp.129-149.

[5] N. Tetsutani,"Stereoscopic Display Method Employing Eye-position Tracking and HDTV LCD-projector," Proc. of Int. Workshop on HDTV '92,1992.

[6] N. Terashima,"A New Concept for Future Telecommunication," Proc. of TAO First Int. Symp. on Three Dimensional Image Communication Technologies,S-4-5,1993.

[7] N. Terashima,"VR and Communication with a Sensation of Reality," Proc. of IEEE Denshi Tokyo,1993 ,pp.21-26, No.32.

[8] N. Terashima,"Virtual Space Teleconferencing System -A Distributed Virtual Environment- ," Proc. of Broad band Islands '94, North-Holland,1994,pp.35-45.

[9] N. Terashima,"Telesensation -Distributed Virtual Reality-Overview and Prospects," IFIP Transactions Technology and Foundations, North-Holland,1994,pp.49-59.

SURFACE MATCHING FOR CORRELATION OF VIRTUAL MODELS: THEORY AND APPLICATION

351298

Roberto CARACCIOLO, Francesco FANTON and Alessandro GASPARETTO

DIMEG - Department of Innovation in Mechanics and Management
University of Padova
Via Venezia 1 - 35131 Padova - ITALY
Tel: ++39 49 828 6857 Fax: ++39 49 828 6816
e-mail: gaspare@hpdimeg.unipd.it

p. 8

ABSTRACT

Virtual reality can enable a robot user to off line generate and test in a virtual environment a sequence of operations to be executed by the robot in an assembly cell. Virtual models of objects are to be correlated to the real entities they represent by means of a suitable transformation. A solution to the correlation problem, which is basically a problem of 3-dimensional adjusting, has been found exploiting the surface matching theory. An iterative algorithm has been developed, which matches the geometric surface representing the shape of the virtual model of an object, with a set of points measured on the surface in the real world. A peculiar feature of the algorithm is to work also if there is no one-to-one correspondence between the measured points and those representing the surface model. Furthermore, the problem of avoiding convergence to local minima is solved, by defining a starting set of states ensuring convergence to the global minimum. The developed algorithm has been tested by simulation. Finally, this paper proposes a specific application, i.e. correlating a robotized cell, equipped for biomedical use, with its virtual representation.

1. INTRODUCTION

The most recent developments of computer graphics allow to create high quality virtual representations of real entities.

Such virtual images provide a useful representation of the real world only if a transformation is defined, correlating the virtual models to the real world.

This paper investigates the problem of finding a correlation between a real entity and its virtual model. Such a problem is often encountered in many specialistic fields (i.e. in biomedical applications).

Virtual reality, intended as the capability to represent a 3-dimensional environment by means of virtual models of the objects constituting it, is used in robotics as a powerful support to off line programming.

As a matter of fact, the off line programming technique increases the productivity of a robotized cell, by avoiding that the robot be stopped for a long time, in order to be reprogrammed by means of teach-in operations.

Recent developments of CAD systems allow to build robotic simulators that can associate the typical CAD data structures with high quality images. These features enable the user of the simulator to off line generate operating sequences representing the movements of the robot and to test its interactions with the parts inside the cell.

These sequences, easily generated in the virtual environment, can be applied to the real cell only if the correlation between the virtual and the real cell is known. A recent paper [1] describes a procedure of 2-dimensional adjusting, that finds this correlation in the case of an object lying on a working table. This procedure has been tested and applied to the field of automatic assembling. An infrared sensor is used to detect the position of the object.

This paper proposes a more general solution of the adjusting problem, i.e. a solution in the 3-dimensional case. The approach to the problem is rather different: a laser sensor has been used instead of the infrared sensor, so that analog distance measurement in a longer range are now possible, and the developed algorithm is based on the surface matching theory instead of simpler 2-dimensional geometric considerations.

The paper is organized in three main sections: the first section contains an overview of the surface matching theory, the second one proposes an *ad hoc* algorithm to solve the surface matching problem in the 3-dimensional case and some tests to validate it; finally, a combined robotic and biomedical application is discussed.

2. THEORY

The surface matching theory is aimed at finding a correlation between two different representations of the same surface.
In most applications one of these representations is obtained scanning the surface of a real object by a sensor, so that all the data are referred to the sensor's reference frame, whereas the other representation is a virtual model of the same surface, stored in the memory of a computer.
A two-level definition of the matching problem can be given, depending on how the surfaces are represented:
1) given two sets of points, representing the same surface in two different reference frames, find the rigid transformation (expressed by a rototranslation matrix) mapping one set of points into the other. Such a transformation has the following characteristics:

a) must be optimal with respect to some criterion (e.g. minimize the maximum or the mean squared difference of the distances between corresponding points);

b) must work for sets of points with different dimensions;

c) must work also if points in one set do not correspond exactly to points in the other;

d) must work also if the points of the data set are corrupted by noise.

2) given a real surface and its virtual model, a set of points is obtained by sampling the real surface. Find the rigid transformation rototranslating the modeled surface in order to minimize its "distance" from the set of points. This case may be seen as a generalization of the previous one: therefore, the transformation must have the same characteristics a) thru d).

2.1 State of the art

The following are some more formal remarks on the matching problem.
Be X a set of points and $(\boldsymbol{R},\vec{t}\,)$ a rototranslation defined by a rotation matrix $\boldsymbol{R}$ and a translation vector $\vec{t}$, let us call P the set of points obtained applying the rototranslation $(\boldsymbol{R},\vec{t}\,)$ to the set X. It is simple to obtain the rototranslation matrix $(\boldsymbol{R},\vec{t}\,)$ starting from the knowledge of X and P, if the one-to-one correspondence of the points of the two sets is known.
The problem of determining the transformation $(\boldsymbol{R},\vec{t}\,)$ becomes more difficult if the points of one set are affected by noise, in the sense that the relationship

$$\vec{x}_i = \mathbf{R}\cdot\vec{p}_i + \vec{t} \tag{1}$$

does not hold for all pairs of points of X and P. In the above equation $\vec{x}_i$ and $\vec{p}_i$ are the coordinates of the i-th point (i = 1...N) of the sets X and P respectively.
In this case the problem becomes a minimization problem: it is required to found the matrix $\boldsymbol{R}$ and the vector $\vec{t}$ that minimize the sum of the errors

$$\vec{e}_i = \vec{x}_i - \mathbf{R}\cdot\vec{p}_i - \vec{t} \tag{2}$$

The general matching problem does not require any one-to-one correspondence between the points of X and the points of P. This implies that no rototranslation exists, which maps exactly every point of X into a point of P even in the case of zero noise.
Some authors have investigated the matching problem, applying their algorithms to specific cases.
If the one-to-one correspondence is known the matching problem can be solved using the methods proposed by Horn and Haralick.
Horn [2] proposes a very simple method to determine the rototranslation matrix in the 2-dimensional case (i.e. when all points of each set lie in the same plane). A 3x4 rototranslation matrix is obtained finding the 3x3 rotation matrix first and then the 3x1 translation vector.
The algorithm proceeds as follows: given two sets of points X and P, with the same dimension N, considering three not aligned points of the P set, and the corresponding ones of the X set, build an adequate reference frame for each set of points, according to the following rules:
a) the origin be the first point;
b) the X axis be the line connecting the first and the second point;
c) the Y axis be the line, coplanar with the three points, and normal to the X axis;
d) the Z axis be chosen following the right hand rule.
Once these frames have been built, it is straightforward to find the 3x3 rotation matrix $\boldsymbol{R}$ between them. The translation vector is then found recalling that corresponding points in the two sets are linked by the following relationship:

$$\vec{x}_i = \mathbf{R}\cdot\vec{p}_i + \vec{t} \tag{3}$$

The algorithm yields an exact result only if the points are not affected by noise; otherwise, it is not possible to find the translation satisfying the above equation exactly, but the transformation correlating each pair of corresponding points is affected by an error:

$$\vec{e}_i = \vec{x}_i - \mathbf{R}\cdot\vec{p}_i - \vec{t} \tag{4}$$

Thus, the problem becomes: find the rigid transformation that minimizes the sum of the squared errors, due to the transformation of all the points of the

P set:

$$\varepsilon^2 = \sum_{i=1}^{N} \|e_i\|^2 \tag{5}$$

Another approach to solve the problem in the 2-dimensional case has been developed by Haralick [3]. His method finds the 3x3 rotation matrix considering all the points in the set simultaneously (whereas the Horn's technique considered only three points at a time). If $N = N_x = N_p$ is the number of points in each of the two sets, the mean squared error ε^2 to be minimized is now:

$$\varepsilon^2 = \sum_{i=1}^{N} w_i \cdot \|\vec{x}_i - (\mathbf{R} \cdot \vec{p}_i + \vec{t})\|^2 \tag{6}$$

where the weights w_i should meet the conditions:

$$w_i \geq 0, \sum_{i=1}^{N} w_i = 1 \tag{7}$$

By choosing the weights in a convenient way, the method is made robust and stable. A good rule to choose the weights is to associate a greater weight to those points with lower squared error. The steps to build the Haralick estimator are the followings:

a) starting from an initial value for the rotation matrix $\boldsymbol{R}$ and the translation vector $\vec{t}$, determine the errors e_i^2 for each pair of corresponding points;

b) the weights can now be chosen, using the Tukey function, applied to the errors e_i^2:

$$w_i = \begin{cases} \left[1 - \dfrac{\|e_i\|^2}{(c \cdot S)^2}\right]^2 & \mathit{if} \|e_i\| \leq c \cdot S \\ 0 & \mathit{otherwise} \end{cases} \tag{8}$$

where c and S are parameters of the Tukey function; namely, c is chosen between 6 and 12, and S is the median of the absolute deviation of the errors e_i^2;

c) solve the minimization problem using the weights that have been computed in the previous step; in this way new values for $\boldsymbol{R}$ and $\vec{t}$ are obtained;

d) iterate the steps b) and c) until the global error ε^2 decreases below a fixed threshold.

The Haralick technique can be extended to the 3-dimensional case (see [3]).

Another solution to the surface matching problem in the 3-dimensional case is given by Besl. He proposes a method, based on the Iterative Closest Point (ICP) algorithm, to match two 3-dimensional surfaces. This technique, described in [4], utilizes quaternions to represent rotations; thus, the rototranslation transformation is described by a 7-dimensional vector instead of a 3x4 matrix. This method reveals itself accurate and computationally efficient; furthermore, it works also if there is no one-to-one correspondence between the two sets of points representing the surfaces.

3. DESCRIPTION OF THE ALGORITHM

An algorithm has been developed, which matches the descriptor of a surface representing a virtual model (e.g. a set of points gotten from the model), with a surface descriptor extracted from the corresponding real object (e.g. a set of points measured on the surface of the object in the real world). This algorithm is a modification and an evolution of the Closest Point Algorithm proposed by Besl [4]. An important feature of this algorithm is that it works also if there is no one-to-one correspondence between the points of the X and the P sets.

Some preliminary definitions will now be given. Let us call X the model set, i.e. a set of points representing the modeled surface and P the data set, i.e. a set of points representing the real surface (e.g. points gotten sampling the surface by means of a sensor). Both sets have the same dimension N.

The matching problem is solved finding:

- a correspondence K between the two sets of points;
- a rotation matrix $\boldsymbol{R}$ and a translation vector $\vec{t}$ linking each point of the model with the corresponding data point, that minimizes the sum of the squared errors (5).

Two kinds of errors that are implicitly included in (5) are: the measurement errors (affecting the data set P), and the errors in the model (affecting the model set X). The latter are due to the fact that the virtual surface is not an exact model of the real surface; the former in most cases may be neglected. However, both these errors cannot be minimized by the matching algorithm.

The matching problem can be classified into:

- global matching
- local matching.

In the first case, there is a biunivocal correspondence between all the points of the model and all the data points, because the data represent the whole surface. It is required to determine $\boldsymbol{R}$ and $\vec{t}$ that minimize the function G:

$$G(X,P) = \min_{(R,\vec{t})} \|X - (\mathbf{R} \cdot P + \vec{t})\| \tag{9}$$

In the case of local matching, the data represent only a part of the surface (thus, the dimension of P is necessarily smaller than the dimension of X). It is required to determine not only R and $\vec{t}$, but also which part Z of the model X minimizes the function L:

$$L(X,P)=\min_{Z\subseteq X}\left[\min_{(R,\vec{t})}\|Z-(\mathbf{R}\cdot P+\vec{t})\|\right]= \\ =\min_{Z\subseteq X}G(Z,P) \qquad (10)$$

Before going further into the description of the algorithm, let us define which type of distance between geometric entities is assumed in the algorithm.
The distance between two points $\vec{r}_1=(x_1,y_1,z_1)$ and $\vec{r}_2=(x_2,y_2,z_2)$ is assumed to be the euclidean distance:

$$d(\vec{r}_1,\vec{r}_2)=\|\vec{r}_1-\vec{r}_2\|= \\ =\sqrt{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2} \qquad (11)$$

Let $A=\{\vec{a}_i\}$ be a set of points, (i=1..N_a), where N_a is the number of points, the distance between a point $\vec{p}$ and the set A is defined as:

$$d(\vec{p},A)=\min_{i\in\{1..N_a\}} d(\vec{p},\vec{a}_i) \qquad (12)$$

Let l be a segment connecting $\vec{r}_1$ and $\vec{r}_2$, the distance between a point $\vec{p}$ and the segment l is:

$$d(\vec{p},l)=\min_{u+v=1} \|u\cdot\vec{r}_1+v\cdot\vec{r}_2-\vec{p}\| \qquad (13)$$

where $u\in[0,1]$ and $v\in[0,1]$.
Then, if $L=\{l_i\}$ is a set of segments, (i=1..N_l), where N_l is the number of segments, the distance between a point $\vec{p}$ and the set L is defined as:

$$d(\vec{p},L)=\min_{i\in\{1..N_l\}} d(\vec{p},l_i) \qquad (14)$$

Let t be a triangle whose vertices are $\vec{r}_1$, $\vec{r}_2$ and $\vec{r}_3$; the distance between a point $\vec{p}$ and the triangle t is defined as:

$$d(\vec{p},t)=\min_{u+v+w=1} \|u\cdot\vec{r}_1+v\cdot\vec{r}_2+w\cdot\vec{r}_3-\vec{p}\| \qquad (15)$$

where $u\in[0,1]$, $v\in[0,1]$ and $w\in[0,1]$.

Then, if $T=\{t_i\}$ is a set of triangles, (i=1..N_t), where N_t is the number of triangles, the distance between a point $\vec{p}$ and the set T is defined as:

$$d(\vec{p},T)=\min_{i\in\{1..N_t\}} d(\vec{p},t_i) \qquad (16)$$

Both a curve and a parametric surface are described by a relationship $\vec{r}(\vec{u})$, where:

$\vec{u}=u\in A\subset\Re^1$ for parametric curves;
$\vec{u}=(u,v)\in A\subset\Re^2$ for parametric surfaces.
The domain A is a segment if $\vec{r}(\vec{u})$ is a curve; it is a closed region in the plane if $\vec{r}(\vec{u})$ is a surface.
The distance between a point $\vec{p}$ and the parametric entity E is defined as:

$$d(\vec{p},E)=\min_{\vec{r}(\vec{u})\in E} d(\vec{p},\vec{r}(\vec{u})) \qquad (17)$$

Then, if $F=\{E_i\}$ is a set of parametric entities, (i=1..N_e), where N_e is the number of entities, the distance between a point $\vec{p}$ and the set F is defined as:

$$d(\vec{p},F)=\min_{i\in\{1..N_e\}} d(\vec{p},\vec{E}_i) \qquad (18)$$

These mathematical concepts will be useful in the following description of the global surface matching algorithm. The subscript k is used to indicate the quantities involved in the k-th iteration of the algorithm.

Let $P=\{\vec{p}_i\}$ and $X=\{\vec{x}_i\}$ be the two sets of points to be matched.
If P and X have the same dimension ($N_x=N_p$), the matching problem can be solved using the above described Haralick method, setting the initial conditions: $\mathbf{R}_0=I_3$, $\vec{t}_0=\vec{0}$ (so that $P_0\equiv P$). We define the Q operator as the function that performs the registration between P and X, i.e. computes the optimal rotation matrix that matches P and X. So, for each iteration new values for R and $\vec{t}$ are obtained by applying the Q operator as follows:

$$(\mathbf{R}_k,\vec{t}_k,d_k)=Q(P_k,X) \quad k>1 \qquad (19)$$

where d_k is the mean squared error given by (5). The value of P_k is obtained applying the rotation $\mathbf{R}_{k-1}$ and the translation $\vec{t}_{k-1}$ to the whole set P_{k-1}, summarized by the formula:

$$P_k=\mathbf{R}_{k-1}P_{k-1}+\vec{t}_{k-1} \qquad (20)$$

The iterations stop when the absolute value of the difference between two consecutive mean squared errors is lower than a fixed positive threshold τ :

$$|d_k - d_{k+1}| < \tau \tag{21}$$

Now, let us consider the more general problem of matching two sets of points with different dimensions. To solve this problem an iterative algorithm of the "closest point" type is used.

Let us suppose that the dimension of the set of the model points is greater than that of the set of data points ($N_x > N_p$), and let us call Y_k the set of the N_p points of X which are the closest to the points of P (i.e. are the "best correspondent points") in the k-th iteration; this defines, for each iteration, a new correspondence K. Let us call C the operator performing this computation:

$$Y_k = C(P_k, X) \tag{22}$$

Now the optimal rotation matrix R and the optimal translation vector $\vec{t}$ can be computed using the above defined Q operator applied to the Y_k set:

$$(\mathbf{R}_k, \vec{t}_k, d_k) = Q(P_k, Y_k) \tag{23}$$

The rototranslation $(\mathbf{R}_k, \vec{t}_k)$ thus computed is then applied to all the points of X, obtaining a new set P_{k+1} which is closer to the X set (see [4] for a proof).

The C operator is now applied to the new set P_{k+1} in order to determine the new set Y_{k+1} of points closest to X.

The loop is iterated until the difference between the mean squared errors in two consecutive iterations is lower than a fixed positive threshold τ.

The convergence of this algorithm to a local minimum has been demonstrated [4]. However, the convergence to the global minimum is not assured in the general case. A way to make the algorithm converge to the smallest local minimum is to start the algorithm choosing $\mathbf{R}_0$ in an adequate set of initial rotations, called "states", instead of choosing $\mathbf{R}_0 = \mathbf{I}_3$. Besl [4] has investigated how to find a suitable set of initial states.

This algorithm can be used also to solve the local matching problem; in this case, it is necessary to introduce a set of initial translations in addition to the set of initial rotations, in order to avoid convergence to a local minimum.

The algorithm has been tested for both global and local matching. The set X of points of the model is made of 55 points sampled on the surface of an ellipsoid. Four tests have been made using different P sets.

The first test applies the algorithm to the case of a global matching in ideal conditions (i.e. there is no noise). Of course, the algorithm converges exactly.

In the second test a gaussian noise has been added, to account for errors in the model and in the measurement. The error has zero mean and its variance is one twentieth of the maximum absolute value of the coordinates of the data points.

The algorithm converges after testing four initial rotations.

Fig. 1 and 2 show the sets before and after the algorithm has been run.

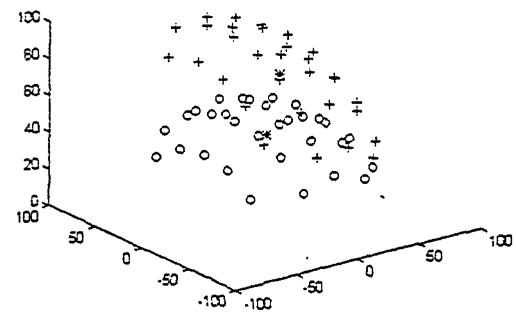


Figure 1 - Sets of points to be matched

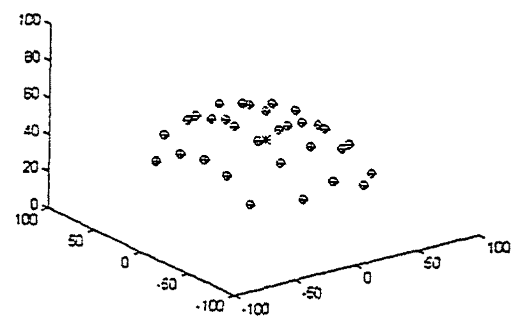


Figure 2 - Result of matching

The third test is a local matching between 24 and 55 points, with additional noise. The results are shown in Fig. 3 and 4.

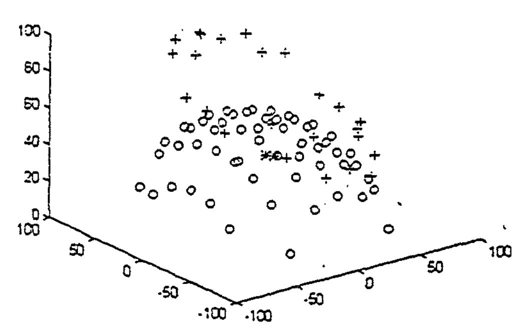


Figure 3 - Sets of points to be matched

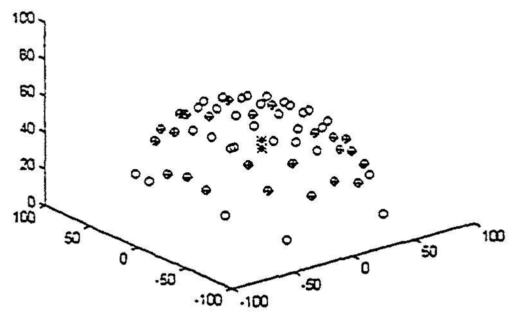


Figure 4 - Result of matching

Finally, the algorithm has been run to match two sets of 15 and 55 points, with a gaussian noise with zero mean and variance three times greater than the previous value. The results are shown in Fig. 5 and 6.

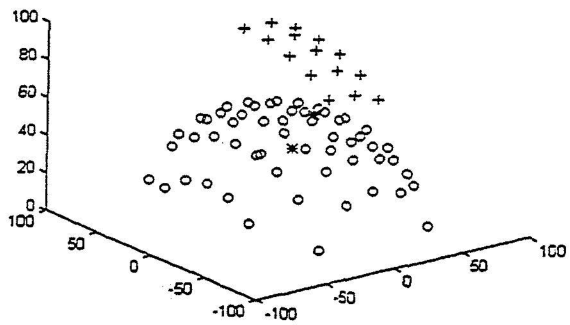


Figure 5 - Sets of points to be matched

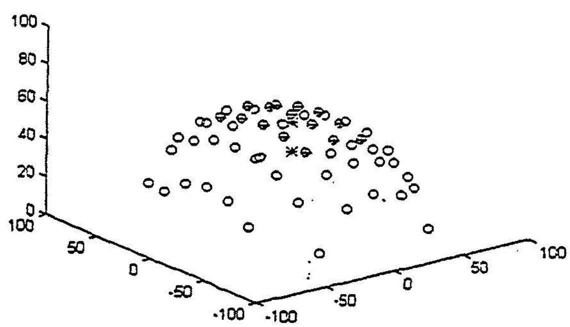


Figure 6 - Result of matching

This technique to match two sets of points can be extended to the case when the model is not represented by a set of points but by a surface. This can be convenient when the surface is expressed in an analytical form (e.g. if the model is built using a CAD system, or if the analytical form of the surface is known). To run the algorithm, the points of the model set are chosen to be the points on the surface which are the nearest to each point of the data set.

In this case the algorithm requires a method to compute the distance between a point and a parametric surface. To compute the distance between a point and a parametric surface, the latter can be approximated by a set of triangles, whose vertices lie on the surface. The shorter the edges of the triangles are, the better the approximation is.

Therefore, the problem of computing the distance between a point and a parametric surface is turned into the problem of computing the distance between a point and a set of triangles, which has been defined above.

The next three figures represent the matching between the set of the data points and the surface expressed in an analytical form.

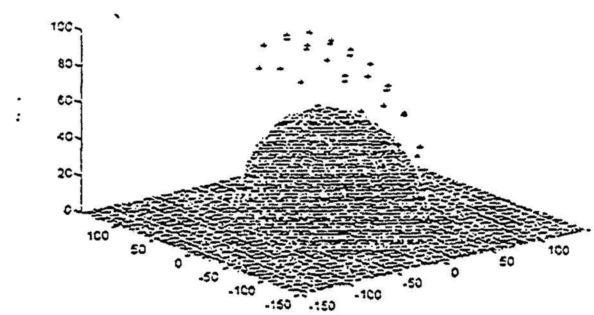


Figure 7 - The surface and the set of points to be matched

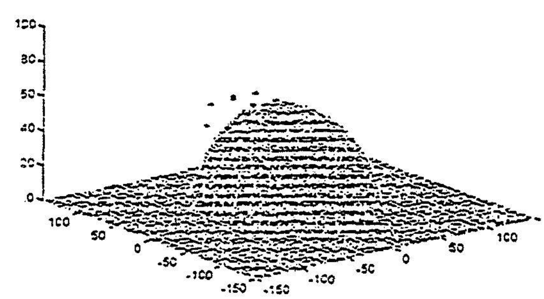


Figure 8 - An intermediate result of matching

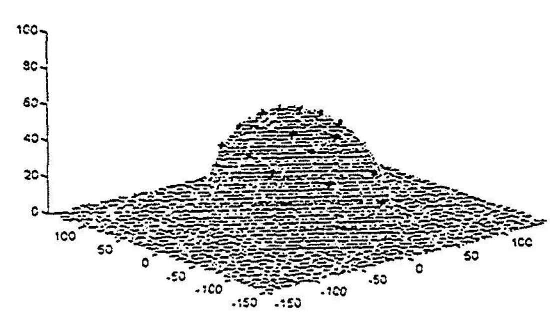


Figure 9 - Result of matching

4. APPLICATION TO ROBOTICS

A special system has been developed, whose structure is made of two levels: a real workcell equipped with an ASEA type IRB2000 industrial 6 d.o.f. robot and a simulation environment based on a prototype of a robotic simulator able to represent exactly the real cell.
This system was initially conceived for automatic assembling purposes; therefore, it has been equipped with a sliding worktable and a set of automatically changeable tools (e.g. grippers, screwers, IR and laser sensors).
The flexibility of the system is such that it can be used also in a rather different research field as robotics applied to biomedics. While in the automatic assembling field a 2-dimensional adjusting procedure turned out to be sufficient in most cases, a more sophisticated 3-dimensional adjusting procedure is necessary to find the real position of the object, being this now a patient.
A major difference from the mechanical case, where all parts are modeled by means of a CAD system which is already integrated with the simulator, is that in the biomedical case the virtual models of the part of interest of the patient are obtained by correlating images gotten by different diagnostical exams (e.g. NMR, CT, DA). (See [6], [7], [10], [14], [15], [19], [20] for reference).
The proposed application refers to a skull represented by a dumb in the real world, and by a virtual model reconstructed from diagnostical images in the simulation environment.
A correlation is then to be established between the virtual model and the patient's skull, by matching the virtual surface and a set of points taken on the real skull using a laser sensor mounted on the end effector of the robot. In this way it is also possible to make a further correlation between the skull and the robot, so that to establish a full correlation between the virtual reality and the real world.
An operating procedure on this skull can then be defined in the simulation environment, and the 3-dimensional surface matching based adjusting procedure can be used to translate the operational sequence into a code suitable for the real cell, and executable by the robot. The translation procedure can be done automatically, since the translator already developed for the robotized assembling system can be employed.

5. CONCLUSIONS

The problem of finding a correlation between a real entity and its virtual model has been investigated in this paper. Solution to this problem can provide a powerful tool in robotics, particularly useful for off line programming.
An algorithm has been proposed, based on the surface matching theory, which matches the surface of a real object with its virtual model. Two cases have been taken into account, namely the matching between two sets of points representing the real and the modeled

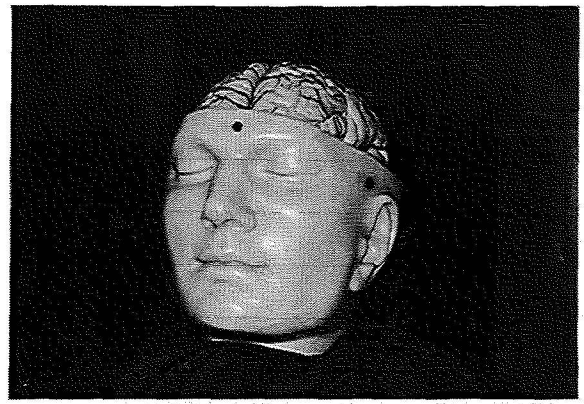


Figure 10 - The real skull

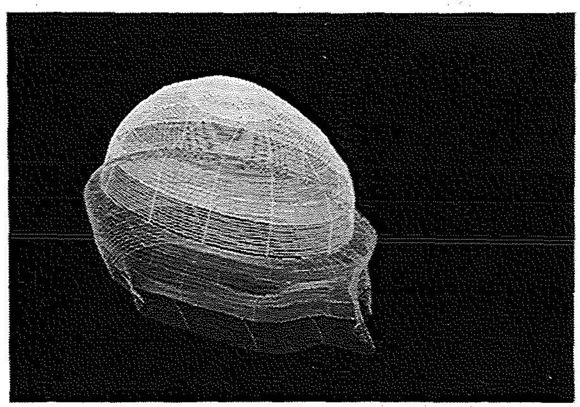


Figure 11 - The CAD model of the skull

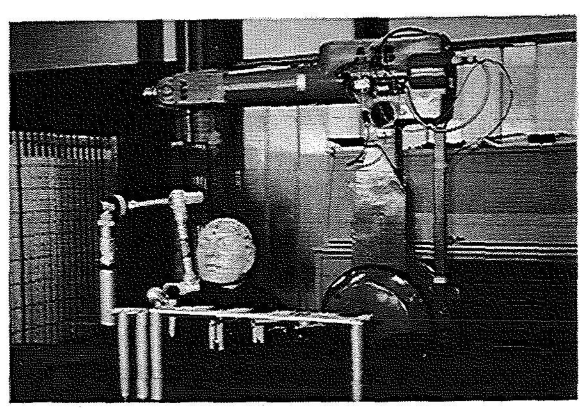


Figure 12 - The real cell

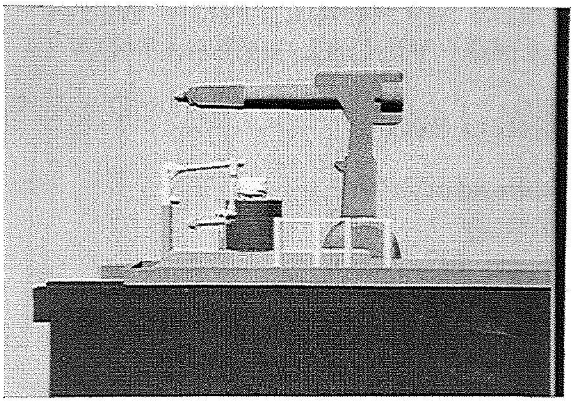


Figure 13 - The virtual model of the cell

surface respectively, and the matching between a set of points gotten from the real surface and the virtual surface, expressed in an analytical form. A peculiar feature of the algorithm is to work also if the two sets of points have different dimensions, and if there is no one-to-one correspondence between them. Moreover, both the global and the local matching problems have been defined and a solution to them has been proposed. The proposed algorithm has been tested by simulation. Finally, a special system, composed of a robotized cell and a simulation environment, initially conceived for automatic assembling purposes, has been presented, and its application to the biomedical field has been discussed.

REFERENCES

[1] S. Badocco, R. Caracciolo, M. Giovagnoni, A. Rossi, "An On-Line Adjusting System for Assembly Robots", Proc. of the Second International Symposium on Measurement and Control in Robotics, Tsukuba Science City, Japan, November 1992.

[2] B. K. P. Horn, "Closed Form Solution of Absolute Orientation Using Unit Quaternion", J. Opt. Soc. Am. A, vol. 4, n. 4, April 1987.

[3] R. M. Haralick et al., "Pose Estimation from Corresponding Point Data", in Machine Vision for Inspection and Measurement, H. Freeman ed., C. E. New York Achademic, 1989.

[4] P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes", IEEE Transactions on Pattern Analysis Intelligence, Vol. 14, n. 2, February 1992.

[5] P. J. Besl, "The Free-Form Surface Matching Problem", in Machine Vision for Three-dimensional Scenes, H. Freeman ed., C. E. New York Achademic, 1990.

[6] N. T. S. Evans, "Combining Imaging Techniques", Clin. Physiol. Meas., vol. 11, Suppl. A, pagg. 97-102, 1990.

[7] R. Dann et al., "Evaluation of Elastic Matching System for Anatomic (CT, MR) and Functional (PET) Cerebral Images", Journal of Computer Assisted Tomography, 13(4): pagg. 603-611, July/August 1989.

[8] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternion", J. Opt. Soc. Am. A, vol.4, n.4, April 1987.

[9] P. T. Fox, J. S. Perlmutter, and M. E. Raichle, " A Stereotactic Method of Anatomical Localization for Positron Emission Tomography", Journal of Computer Assisted Tomography, n.1, January/February 1985, Raven Press, New York.

[10] T. M. Peters, J. A. Clark, et al., "Integrated Stereotaxic Imaging with CT, MR Imaging, and Digital Subtraction Angiography", Radiology , 1986; 161: 821-826.

[11] D. J. Hawkes, D. L. G. Hill, et al.,"Preliminary Work on Interpretation of SPECT Images with the Aid of Registered MR Images and MR Derived 3D Neuro-anatomical Atlas", NATO ASI Series, Vol. F60, 3D Imaging in Medicine, 1990, Edited by K.H.Hohne et al.

[12] B. J. Erickson and C. R. Jack, "Correlation of Single Photon Emission CT with MR Image Data Using Fiduciary Markers", AJNR 14:713-720, May/June 1993.

[13] P. Clarysse, D. Gibon et al., "A Computer-Assisted System for 3-D Frameless Localization in stereotaxic MRI", IEEE Transactions on Medical Imaging, vol.10, n.4, December 1991.

[14] D. N. Levin, "MR and PET data merge in 3-D images of brain", Diagnostic Imaging International, 28:33, January/February 1990.

[15] C. A. Pelizzari, G. T. Y. Chen et al., "Accurate Three-Dimensional Registration of CT, PET, and/or MR Images of the brain", Journal of Computer Assisted Tomography, 13(1): 20-26, January/ February, 1989 Raven Press, Ltd., New York.

[16] C. A. Pelizzari, G. T. Y. Chen, "Image Correlation Techniques in Radiation Therapy Treatment Planning", Computerized Medical Imaging and Graphics, vol. 13, n. 3, 235:240, 1989, Printed in the U.S.A.

[17] B. L. Holman, R. E. Zimmerman et al., "Computer-Assisted Superimposition of Magnetic Resonance and High-Resolution Technetium-99m-HMPAO and Thallium-201 SPECT Images of the Brain", The Journal of Nuclear Medicine, vol.32, n.8, August 1991.

[18] D. N. Levin, C. A. Pelizzari et al., "Retrospective Geometric Correlation of MR, CT, and PET Images", Radiology 1988; 169: 817-823.

[19] W. Zhao and T. Y. Young, "A new approach to three-dimensional reconstruction with application to autoradiographic images", Computers in Medicine, vol. 13, n. 3, 1991.

[20] M. L. Kessler, S. Pitluck et al., "Integration of Multimodality Imaging Data for Radioterapy Treatment Planning", Int. J. Radiation Oncology Biol. Phys., vol.21, 1991, pp. 1653-1667.

[21] S. Lavallée, P. Cinquin et al., "IGOR: Image Guided Operating Robot. Metodology, medical applications, results", Innov. Tech. Biol. Med.,vol.13, No 4,1992.

[22] D. Glauser, P. Flury, "Conception of a Robot Dedicated to Neurosurgical Operations", Proc. of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1991.

[23] Y. S. Kwoh, J. Huo et al., "A Robot with Improved Absolute Positioning Accuracy for CT Guided Stereotactic Brain Surgery", IEEE Transaction on Biomed. Eng., vol.35, 153-160, 1988.

1995109569

N95- 15984

A VIRTUAL WORK SPACE FOR BOTH HANDS MANIPULATION WITH COHERENCY BETWEEN KINESTHETIC AND VISUAL SENSATION

351303

P. 7

Masahiro Ishii, P.Sukanya, Makoto Sato

Precision and Intelligence Laboratory
Tokyo Institute of Technology
Nagatsuda 4259, Midoriku, Yokohama 227, Japan
Tel: 81-45-924-5050, Fax: 81-45-921-0898
E-mail: {mishii, sukanya, msato}@pi.titech.ac.jp

ABSTRACT

This paper is about constructing a virtual work space for performing any tasks by both hands manipulation. We intend to provide a virtual environment that can encourage users to accomplish any tasks as they usually act in the real environment. Our approach is using a three dimensional spatial interface device that allows the user to handle virtual objects directly by free hands and be feel-able some physical properties of the virtual objects such as contact, weight, etc. We have investigated the suitable conditions for constructing our virtual work space by simulating some basic assembly work, a Face-and-Fit task. Then select the conditions that the subjects feel most comfortable in performing this task to set up our virtual work space. Finally, we have verified the possibility to perform more complex tasks in this virtual work space by providing some simple virtual models then let subjects create new models by assembling these component models together. The subjects can naturally perform assembly operations and accomplish the task. Our evaluation shows that this virtual work space has potential to be used for performing any tasks that need hands manipulation or cooperation between both hands in natural manner.

KEYWORDS: Virtual reality, 3D modeling, cooperation between both hands, multi-modalities

INTRODUCTION

Recently, many three dimensional (3D) spatial interface devices have been proposed. However, each is appropriate for each kind of work[1][2][6][7]. Now, we still lack of the interface device that can immerse the user into the virtual work space, then allows him/her to perform any tasks as desire. For example, to create a new 3D model, it allows the user to grip, rotate or twist virtual models at any orientations arbitrarily by hands manipulation directly. To construct such the virtual work space, it is indispensable to consider the effective interaction communication between human and machine[8][9]. The multi-modal system is a concept which we adopt to provide information to the user in multi-sensory channels simultaneously as we usually get information in the real environment. Primarily, sight and touch are the senses that we have utilized. We use SPIDAR (SPace Interface Device for Artificial Reality) as the 3D spatial interface device to construct such the virtual work space. SPIDAR has been previously proposed by M.Sato et al.[3][4].

On the other hand, let us mention methods for forming a 3D model manually as we practise in the real environment. There are two basic methods:- Extraction method and Combination method[5].

- *Extraction method*
  It is the method that a new model is created by deforming the original model

- *Combination method*
  It is the method that a new model is created by combining one model with another model in an arbitrary orientation

Actually, to form a new model both by extraction method and by combination method, we are familiar with manipulation by cooperative between both hands. So the interface device that has a capacity of handle by both hands is needed to per-

C-2

form this task in the virtual work space.

Originally, SPIDAR has been developed for single hand manipulation. Since some kinds of work are performed by cooperative between both hands more effective than by single hand, here we have enhanced it for both hands manipulation.

This paper aims to construct a virtual work space primarily for performing a task such as forming a 3D model manually such mentioned above. Finally, we have evaluated it by simulating an environment for forming a 3D model by combination method. Then let subjects perform this task.

OVERVIEW OF SPIDAR

SPIDAR is a 3D spatial interface device previously proposed by M.Sato et al. Originally, it was developed for single hand manipulation. The general structure of it is shown in Figure 1. To perform a task with this interface, the user needs only to put his/her thumb and index finger into the provided caps. Each cap is held by four strings that are wound around pulleys attached with electrical motors at each corner of the cubic frame. The user can move both the thumb and the index finger arbitrarily. The motion of each finger that is the motion of each cap also is detected by the rotary encoders attached with the motors, so the position of each finger can be caculated[4]. By controlling the tension of the strings, we can provide force sensations to the user via the caps. Moreover, we can vary both the magnitude and the direction of forces arbitrarily in the range from 0N to 4N with a step of 0.016 N [4].

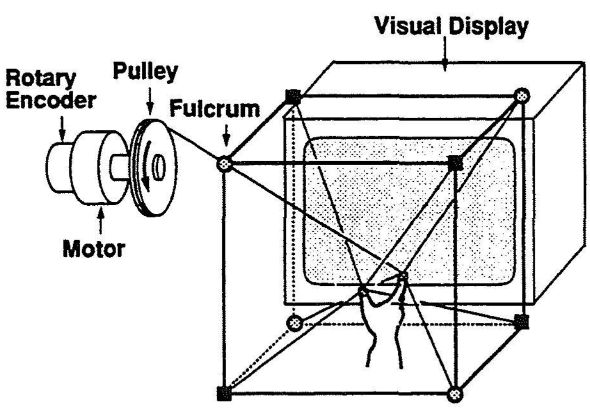


Figure 1 SPIDAR for single hand manipulation

Figure 2 shows the range of motion of the thumb and the index finger that applying force-feedback sensation is effective[3]. Each space for each finger is the tetrahedron whose vertices are the four fulcrums.

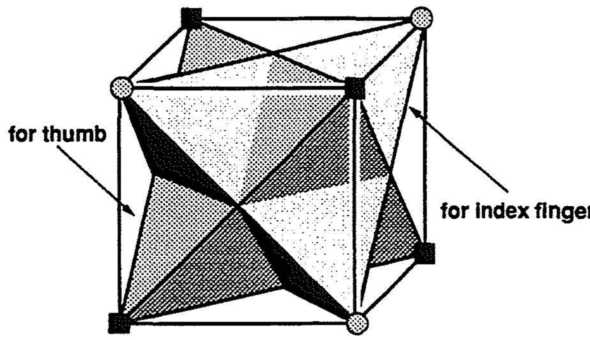


Figure 2 Range of motion of the thumb and the index finger

SPIDAR is a 3D interface device that can track fingers' position and generate force-feedback sensation to the user during manipulating virtual objects. It allows the user to touch and handle virtual objects directly by free hand.

SUITABLE VIRTUAL ENVIRONMENT INVESTIGATION

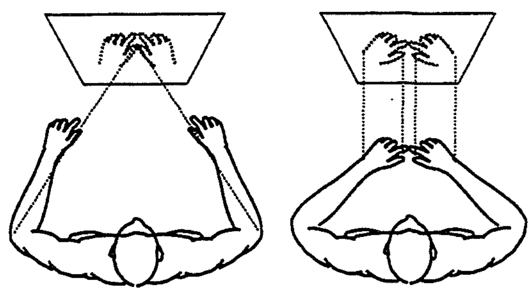
Coherency Between Kinesthetic and Visual Sensation

In our virtual work space, we use the interface device, SPIDAR, which the user has to communicate with the machine by hand movements controlling away from the display screen surface, so we must consider a natural act of eye-hand coordination to make information coincide. Here, our approach is reflecting the images of the thumbs and the index fingers on the display screen then apply force-feedback sensation and change the poses of the virtual objects corresponding to the situations of the images of fingers relative to the virtual objects. We have considered 2 methods for reflecting the images of fingers on the display screen as follows:

- *Lengthened arms method*
  This method reflects the images of fingers by

assuming that the user's arms are lengthened from the current position to the display screen (see Figure 3a). We consider particularly how far the distance between the left and the right hands should be to make the images of fingers of both hands be reflected as if they just touch each other. Let us call *latent distance.* We search for the suitable latent distance by an experiment detailed in Experimental Study section.

- *Projected hands method*
  By this method, the images of fingers are reflected parallel to the actual position (See Figure 3b),so the latent distance is zero.



(a) lengthened arms (b) projected hands

Figure 3 Methods for reflecting the images of fingers on the display screen

We determine which method is suitable for coherency between kinesthetic and visual sensation by an experimental study detailed later.

For providing visual information, we use computer graphic display system for generating stereoscopic images that the user needs to wear stereoscopic glasses to perceive 3D perspective view of the images. We currently do not plan to use head-mounted displays (HMDs) since we feel the current HMD technology is too encumbering and of too limited resolution for viewing complex data.

SPIDAR for Both Hands Manipulation

As mentioned in the previous section, originally, SPIDAR has been developed for single hand manipulation. To enhance it for both hands manipulation, we have considered various styles of setting strings. Some examples of new structures are shown in Figure 4, where the circles are the rough boundaries of each hand motion.

Figure 4a is the structure that is constructed by connecting SPIDAR for single hand manipulation 2 sets together. In this case, the problem of the interference of strings rarely occurs. However, the positions of the left and the right hands are separated rather far, so it is difficult to perform some operations that need the cooperation between both hands in the near distance in the real environment and it can not use with reflecting the images of fingers by projected hands method. On the other hand, the structures shown in Figure 4b and 4c can be used with reflecting the images of fingers both by lengthened arms method and by projected hands method, but the interference of strings occurs more often than the structure shown in Figure 4a.

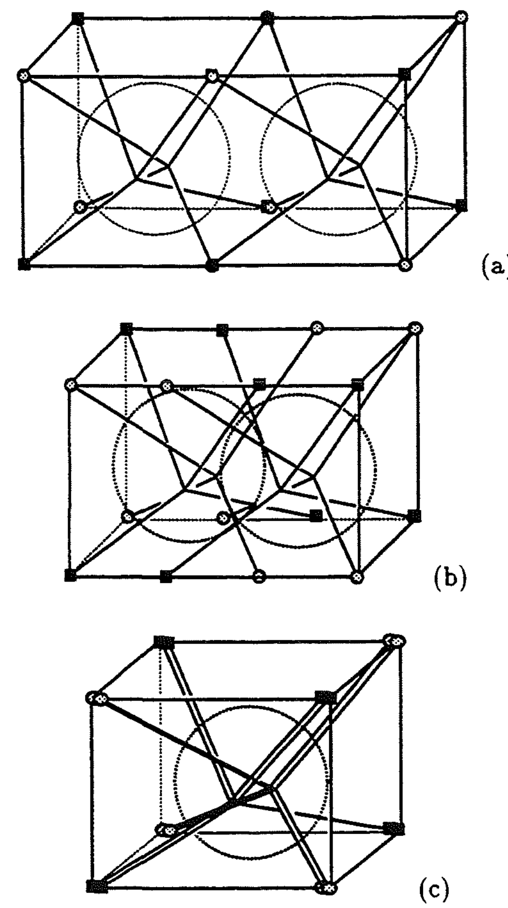


Figure 4 Examples of setting strings of SPIDAR for both hands manipulation

Experimental Study

From the conditions that we have considered above, we determine which ones are the suitable conditions by experimental study. In the experiment, we simulate an environment for performing a Face-and-Fit task (pick up two objects from a number of provided objects, then turn them until the desired sides face each other and unite them). The virtual target objects are two cubic cubes (size 40x40x40 mm^3, weight 50 g) and the initial distance between them is 120 mm. We use 9 styles of SPIDAR structure with reflecting the images of fingers by lengthened arms method varying 8 values of the latent distance: 10, 20, 30, 40, 50, 60, 70, 80 cm. For projected hands method ,which the latent distance is zero, we experiment with the SPIDAR structure shown in Figure 4b. The computer graphic display system is used for generating a real time image with screen updated rate 10 times per second. Force-feedback sensation is generated with force-feedback updated rate 30 times per second. The subject has to wear the provided stereoscopic glasses to see virtual objects as 3D objects. The distance from the subject's eyes to the virtual objects on the display screen is 75 cm. Before the subjects do this experiment, they have been trained until they have enough skills to use this interface device. After each subject finishes the experiment, we have interviewed him/her to collect information about conditions that he/she satisfies in performing the Face-and-Fit task in this environment. Figure 5 is a scene of the experiment.

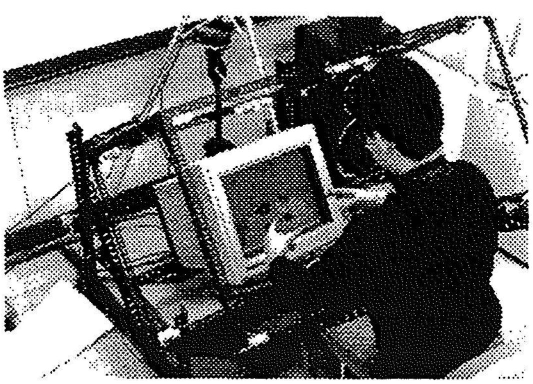
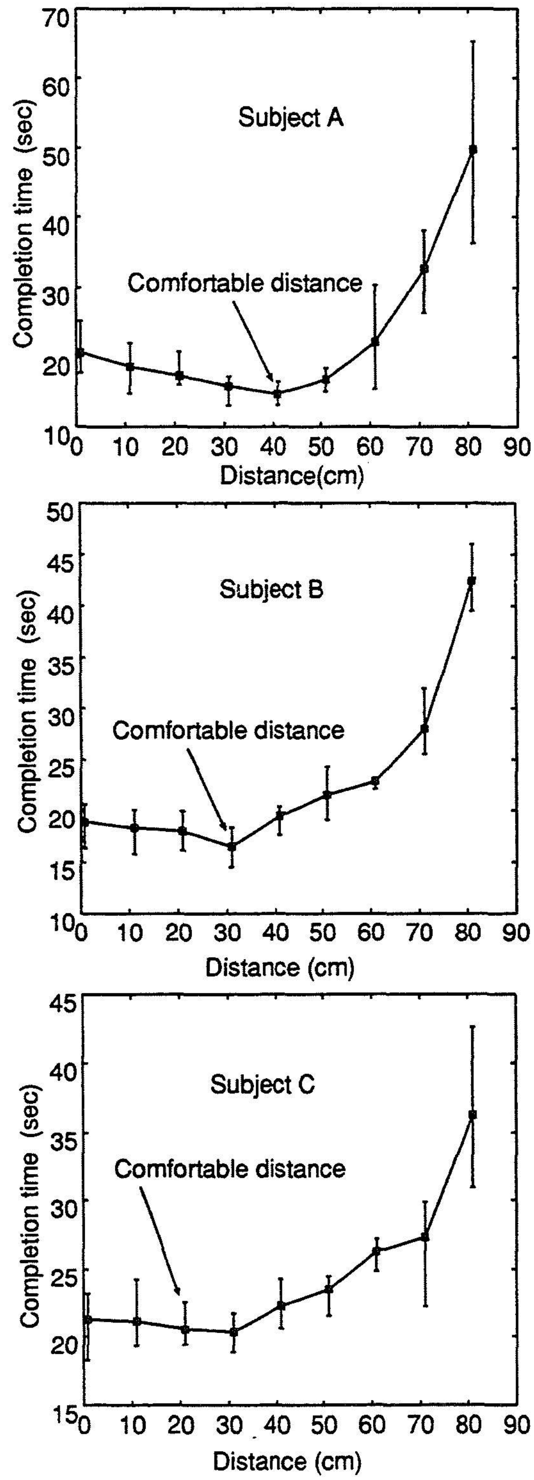


Figure 5 A scene of the experiment

Results and Considerations

Figure 6 is the result that shows the relation between the latent distance and the task completion time of four subjects. The point that is pointed by the arrow is the latent distance that the subject feels most comfortable in performing the task.



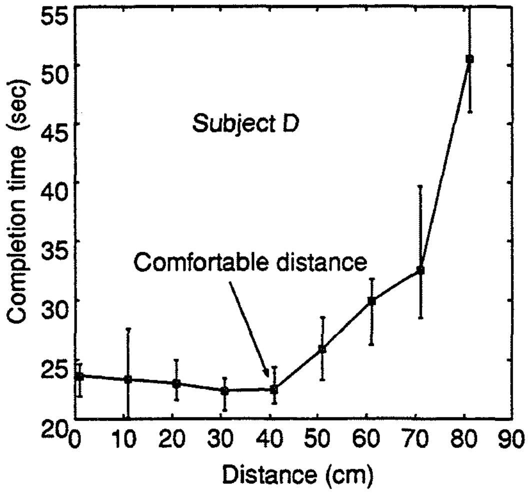


Figure 6 The relation between the latent distance and the task completion time

Subject A can accomplish the task fastest when the latent distance is 40 cm and it is the latent distance that subject A feels most comfortable in performing the task. Subject B can accomplish the task fastest when the latent distance is 30 cm and same as subject A it is the latent distance that subject B feels most comfortable in performing the task. Subject C can accomplish the task fastest when the latent distance is 30 cm, and the latent distance that subject C feels most comfortable in performing the task is 20 cm. Subject D can accomplish the task fastest when the latent distance is 30 cm, and the latent distance that subject D feels most comfortable in performing the task is 40 cm. Although the latent distance that subject C and subject D accomplish the task fastest, and the latent distance that they feel most comfortable in performing the task are different. They are not the discrepant results because when the latent distances are 20 cm and 30 cm for subject C, and 30 cm and 40 cm for subject D, they take time for accomplishing the task in the vicinity.

The most suitable latent distance is 30-40 cm that both makes the users comfortable in performing the task and helps the users to accomplish the task most effective.

Let us consider the latent distance again. Refer to Figure 7, we can establish the formula for calculating the latent distance (FlFr) as follows:

$$\mathrm{FlFr} : \mathrm{ElEr} = \mathrm{VcFr} : \mathrm{VcEr}$$

$$\begin{aligned}\mathrm{FlFr} &= \frac{\mathrm{VcFr} \times \mathrm{ElEr}}{\mathrm{VcEr}} \\ &= \frac{(\mathrm{VcEr} - \mathrm{ErFr}) \times \mathrm{ElEr}}{\mathrm{VcEr}} \\ &= \mathrm{ElEr} - \frac{\mathrm{ErFr} \times \mathrm{ElEr}}{\sqrt{\mathrm{VcH}^2 + (\mathrm{ElEr}/2)^2}} \end{aligned} \qquad (1)$$

In the experiment, the distance from the subject's eyes to the display screen is 75 cm (VcH = 75 cm). The average distance from the elbow to the thumb of four subjects is about 40 cm (ErFr = ElFl = 40 cm) and the average distance from the left elbow to the right elbow is 60 cm (ElEr = 60 cm). By substituting these values in equation (1), we obtain the latent distance value that is about 30 cm corresponding to the result from our experiment.

From these results, it indicates that lengthened arms method is better than projected hands method.

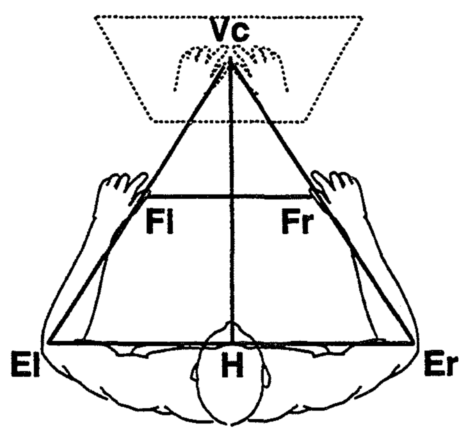
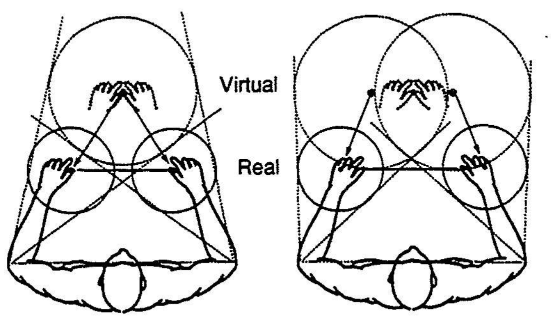


Figure 7 Geometric relation of perceptual information

VIRTUAL WORK SPACE CONSTRUCTION

Setting up the Virtual Work Space

We adopt the suitable conditions that are considered in the previous section to construct our virtual work space. Although it is suggested that the suitable latent distance is 30-40 cm, we have to consider additionally about the boundary of the images of fingers that should be reflected on the display screen. Figure 8 shows two types for defining the boundaries of work spaces of the left and the right hands on the display screen. Type-(a), the boundaries of work spaces of the left and the right hands are located in the same area. Type-(b) the boundaries of work spaces of the left and the right hands are separated but some parts join each other at the center of the total frame. The total boundary of type-(b) is larger than that of type-(a) but type-(b) can not support some operations such as moving the right hand to the leftmost of the total boundary or moving the left hand to the rightmost of the total boundary. However, in performing any tasks, we hardly operate by crossing hands and the larger boundary can present more information to the user, so we select type-(b) for defining the boundaries of work spaces on the display screen.



(a) same work space (b) separate work spaces

Figure 8 Boundaries of work spaces of both hands on the display screen

From the selected conditions, we have constructed a virtual work space for both hands manipulation which has the rough structure as shown in Figure 9.

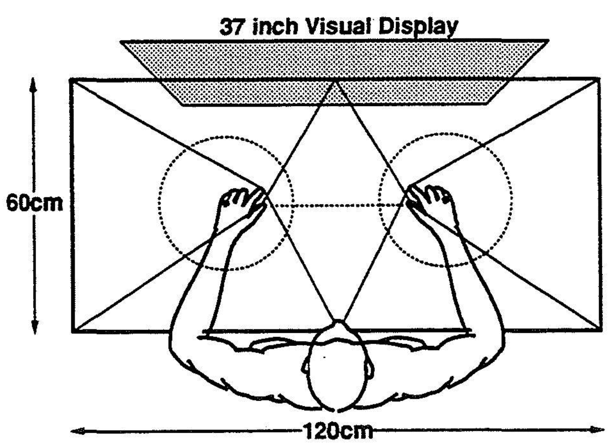


Figure 9 Rough structure of a virtual work space for both hands manipulation

Initial Application

We have evaluated our virtual work space by simulating an environment for forming a 3D model manually by combination method. We provide some simple 3D virtual models such as a sphere, a rod, etc, then let the users create new models by assembling these component models together. The users can perform naturally assembly operations and accomplish this task. Figure 10 is an example that shows some situations of the virtual models during a user is performing this task.

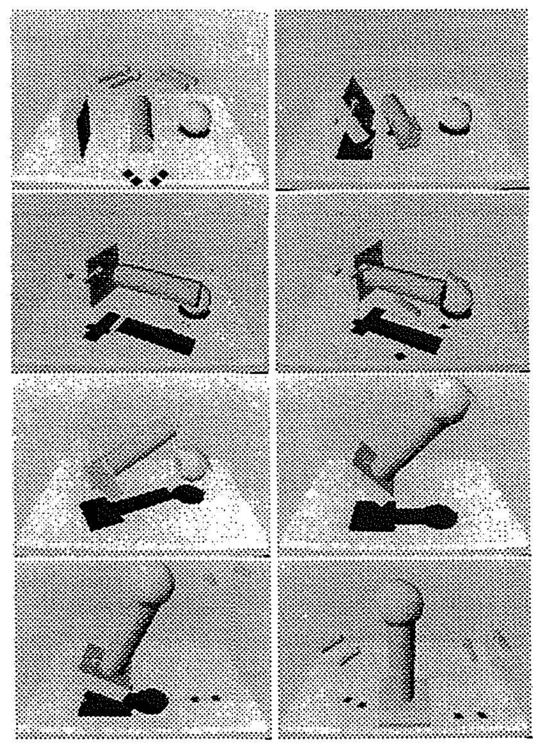


Figure 10 Example of some situations of the virtual models during a user forming a 3D model by combination method

CONCLUSION

In this paper, we have considered the suitable conditions for setting up a comfortable virtual work space for both hands manipulation. Since we use the interface device that communicates with the machine by hand movements controlling away from the display screen surface, the congruity of hand movements and visual information must be considered. In addition to force-feedback sensation, we have proposed reflecting the images of fingers on the display screen to help users to perceive the situations of fingers relative to virtual objects more clearly. Lengthened arms and projected hands are two methods that we have considered and compared by an experiment. The result shows that lengthened arms method is better. We have constructed our virtual work space according to this result then let subjects perform some simulated assembly work. Our evalution shows that this virtual work space has sufficient conditions for supporting users in performing any tasks that need hands manipulation directly or cooperation between both hands in natural manner such as creating a 3D model by assembling the provided component models together. We intend to enhance it for performing more delicate work and plan to utilize auditory sensory to provide supplemental information to the user in the future.

REFERENCES

1. Brooks,F.P., Ouh-Young,M., Batter,J.J. and Kilpatrick,P.J.: "project GROPE - Haptic display for scientific visualization", Computer Graphics, 24,4,1990,pp.177-185.

2. Hinckley,K., Pausch,R., Gobel,J., Kassell, N.:"Passive Real-World Interface Props for Neurosurgical Visualization", ACM CHI'94 Conference on Human Factors in Computing Systems, 1994, pp.452-458.

3. Hirata,Y. and Sato,M.: "3 dimensional interface device for virtual work space", Proceedings of the 1992 IEEE/RSJ International Conference on IROS, 2,1992,pp.889-896.

4. Ishii,M. and Sato,M.: "A 3D Spatial Interface Device Using Tensed Strings", PRESENCE, 3,1,1994,pp.81-86.

5. Isshiki Hachiro:"The world of hand", Education Publication,1980 (In Japanese).

6. Iwata,H.: "Artificial Reality with Force-Feedback: Development of Desktop Virtual Space with Compact Master Manipulator", Computer Graphics, 24,4,1990,pp.165-170.

7. Kabbash,P.,Buxton,W.,Sellen,A.:"Two-Handed Input in a Compound Task", ACM CHI'94 Conference on Human Factors in Computing Systems,1994,pp.417-423.

8. Minsky,M., Ouh-Young,M., Steele,O., Brooks,F.P., and Behensky,M.: "Feeling and seeing: Issues in force display", Computer Graphics, 24,2,1990,pp. 235-244.

9. Yanagida,Y. and Tachi,S.: "Virtual Reality System with Coherent Kinesthetic and Visual Sensation of Presence", Proceedings of 1993 JSME International Conference on Advanced Mechatronics (ICAM),1993,pp.98-103.

Session 5
Sensory Feedback

MASTER-SLAVE SYSTEM WITH FORCE FEEDBACK BASED ON DYNAMICS OF VIRTUAL MODEL

Shuji NOJIMA, Hideki HASHIMOTO

Institute of Industrial Science, University of Tokyo
7-22-1 Roppongi, Minato-ku, Tokyo 106, Japan
Tel +81-3-3479-2766 Fax +81-3-3423-1484
E-mail: nojima@ics.iis.u-tokyo.ac.jp

Abstract

A master-slave system can extend manipulating and sensing capability of a human operator to separated environment from him or her. But the master-slave system has the two serious problems: one is the mechanically large impedance of the system, the other is mechanical complexity of the slave for complex remote tasks. These two problems reduce the efficiency of the remote task through the master-slave system.

If the slave has local intelligence, the slave can help the human operator by using its good points like fast calculation and large memory. After the authors suppose that the slave is the dextrous hand with many degrees-of-freedom (DOF) and it manipulates the object with known shape, it is suggest that the dimensions of the remote work space should be shared by the human operator with the slave.

The effect of the large impedance of the system can be reduced the virtual model, which is a physical model constructed in a computer and has physical parameters as if it was in real world. The way to determine the damping parameter dynamically of the virtual model in one DOF master-slave system is proposed. The experimental result shows that this virtual model is better than the virtual model with fixed damping.

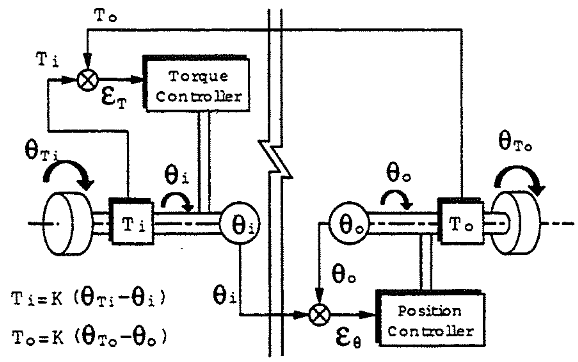


Figure 1: Traditional bilateral master-slave system : position and force control.

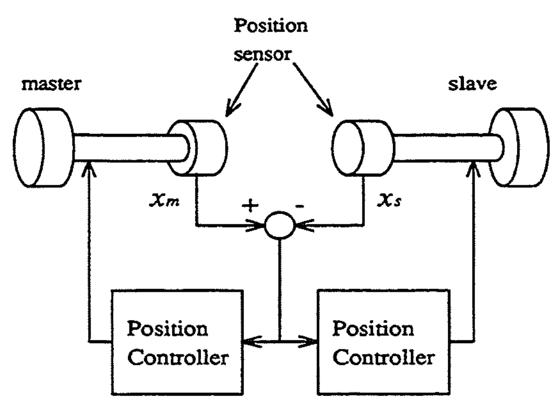


Figure 2: Traditional bilateral master-slave system : position and position control.

1 Introduction

Traditional master-slave systems began in 1940's as a teleoperator through which the human operator could handle the radioactive materials while he or she was separated from that material physically. The traditional master-slave system consisted of the two system, the master arm, and the slave arm [1], [2]. The two systems are connected each other directly by a servo mechanism, called bilateral servo (Figure 1, 2). These traditional systems had two problems:

1. The human operator felt the dynamics of the system in addition to that of the remote environment. Because master-slave systems have mechanically large impedance, the human operator and the remote environment in the system cannot actuate each other accurately.

2. The master arm had the same form as the slave has. When the slave has many degrees-of-freedom (DOF) to perform the remote task dextrously, the human operator must control all the DOF dextrously. This system also needed the wide bandwidth telecommunication between the master and the slave.

The work [3] coped with the first problem, putting an impedance matrix, which changes the impedance of master-slave systems arbitrary, between master and slave. Furuta *et al.* [7] propose the Virtual Internal Model Following Control in order to change dynamics of master slave systems. The discussion in [4] shows *Supervisory Control*, the way to avoid the second problem.

In order to cope with the second problem, this paper suggest that the human operator shares the dimensions of the remote work space with the master-slave system. The authors include the virtual model with physical parameters into the system.

The authors also propose to determine the physical parameters of the virtual model dynamically, and show the way to determine the damping parameter of the virtual model, which stabilize the one DOF master-slave system. The advantage of this parameter determination are shown through one DOF master-slave experiments.

2 Master-Slave System with Virtual Model

In the systems showed in Figure 1 and 2, The master make its position (or torque) coincide with position (or torque) of the slave by this mechanism, and the slave also make its position (or torque) coincide with the position (or torque) of the master by the same mechanism.

These systems have a serious problem. The two systems have mechanically large impedance because the master must actuate a human operator and the slave must produce strong force like human beings to carry out some tasks in a remote environment instead of the operator. Measured force or motion in master and slave system involves this large impedance of the one system or both [4]. These master-slave system cannot accurately communicate force and motion information each other.

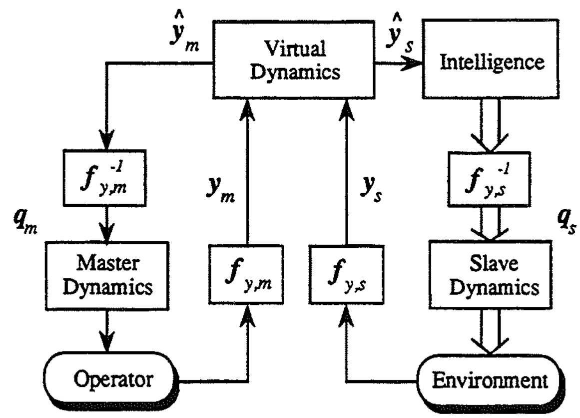


Figure 3: Master slave system with virtual model.

The human operator must feel the force information from the remote environment through the systems in order to perform a remote task as if he or she was in the environment. But that impedance reduces reality about force feeling and makes the remote task inefficient.

The authors include the virtual model in order to change the dynamics of the master-slave system (Figure 3). The virtual model connects the master and slave: it gives the position (or force) reference to these systems while it is given the force (or position) inputs for calculation of its position (or force) by them. The parameters of the virtual model are chosen as they reform the system and keep the stability of it.

3 Dextrous Slave Manipulator

In this section, the authors apply the virtual model to a master-slave system which consists of a dextrous slave manipulator and a master which has less DOF than the slave.

The dextrous slave manipulator is useful to make it perform tasks in remote environment by itself because of its large DOF. This slave can be controlled by a master manipulator with the same form as the slave has. In this case, a human operator of this master-slave system controls all the DOF of the slave by controlling the master. The more DOF the slave and master has, the more they become mechanically heavy, large and uncontrollable for human beings.

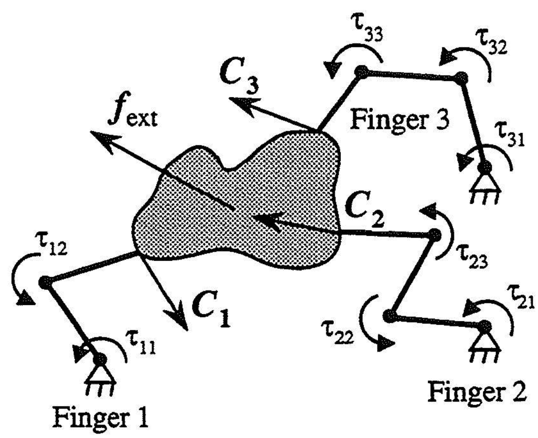


Figure 4: Object grasped by 3 fingers.

If the slave manipulator has a certain intelligence, the human operator and the slave can share dimensions of remote work space, and the slave can be controlled by a master with less DOF than the master.

For example, when the dextrous slave manipulator rotates a valve in a pipeline by grasping, the slave can possess shape and material information of the valve and use a computer as the intelligence. This intelligent slave manipulator makes up for some parts of dimensions of the remote work space, because the slave can autonomously keep a sufficiently grasping force to rotate the valve and vary its configuration with the rotational angle of the grasped valve. The human operator governs only one dimension about the rotation around the center of the valve directly. He or she does not need any feedback information except that of this dimension in order to control the slave and feel the reality of the remote task. Then the master manipulator can reduce its DOF in proportion to work of the slave intelligence.

The dextrous and intelligent slave manipulator reduces the DOF of the master, but dimensions for which the slave can make up change according to remote tasks. A controller of higher level must determine what dimensions the slave can handle autonomously. This section supposes that these dimensions are known.

3.1 Dextrous Manipulation

When m fingers grasp an object with known shape(Figure 4), an external force added to the object and contact forces of the finger tips are related as

$$\boldsymbol{f}_{ext} = \boldsymbol{W}\boldsymbol{c} \tag{1}$$

where $\boldsymbol{f}_{ext} \in \mathbb{R}^6$ is the generalized force vector

$$\boldsymbol{f}_{ext} = [f_x, f_y, f_z, m_x, m_y, m_z]^T = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{m} \end{bmatrix} \tag{2}$$

and $\boldsymbol{c} \in \mathbb{R}^n$ means the contact force vector of the finger tips. The elements of $\boldsymbol{c}$ use *Wrench Representation* [5]. In this representation, each particular type of contact— point contact or soft finger contact, a contact with friction or without, etc. [5]—has a fixed coordinate. The wrench representation treats forces and moments, which are scalar intensities along an axis of the coordinate, as general entity, the *wrench*. n depends on the type and the number of the contact.

The matrix $\boldsymbol{W} \in \mathbb{R}^{6\times n}$ contains the n contact wrench directions in its column. The size, magnitude and rank of the matrix $\boldsymbol{W}$ can vary with changes of the type, variation of the direction and displacement of the position of the contacts. We always assume $\text{rank}(\boldsymbol{W}) = 6$.

The stable grasping needs internal force, which is made by some wrenches exerted by the fingers in addition to minimum degrees of freedom to determine the position and orientation of the object. $\boldsymbol{c}$ must have n dimensions which is grater than six to produce the internal force. It can be split as

$$\boldsymbol{c} = \boldsymbol{c}_p + \boldsymbol{c}_h \tag{3}$$

where $\boldsymbol{c}_p$ is particular solution of the equation (1) and $\boldsymbol{c}_h$ is homogeneous solution of it:

$$\boldsymbol{W}\boldsymbol{c}_h = \boldsymbol{0} \tag{4}$$

The equation (1) has a nontrivial solution because of $n > 6$ and $\text{rank}(\boldsymbol{W}) = 6$. The authors choose $\boldsymbol{c}_p$,

$$\boldsymbol{c}_p = \boldsymbol{W}^+\boldsymbol{f}_{ext} \tag{5}$$

as the solution. $\boldsymbol{W}^+$ is the generalized inverse matrix of $\boldsymbol{W}$ and given as

$$\boldsymbol{W}^+ = \boldsymbol{W}^T[\boldsymbol{W}\boldsymbol{W}^T]^{-1} \tag{6}$$

The authors include the linear mapping N to define the internal force vector explicitly [6].

Definition 1 *The matrix $N \in \mathbb{R}^{n\times(n-6)}$ contains the orthonormal basis vectors $\boldsymbol{c}_{i,h}$ in its columns, which span the $(n-6)$ dimensional null space of $\boldsymbol{W}$.*

$$\boldsymbol{N} = [\boldsymbol{c}_{1,h}, \boldsymbol{c}_{2,h}, \ldots, \boldsymbol{c}_{n-6,h}] \tag{7}$$

N is the linear mapping from $\boldsymbol{f}_{ext}$ in the coordinates of the object to the $\boldsymbol{c}_h$ in the coordinates of the wrench system:

$$\boldsymbol{c}_h = \boldsymbol{N}\boldsymbol{f}_{int} \tag{8}$$

$$f_{int} = N^T c_h \tag{9}$$

The two equations, (1) and (8) can be written into one equation [5].

$$\mathcal{F} = (G^T)^{-1} c \tag{10}$$

$$c = G^T \mathcal{F} \tag{11}$$

A generalized force vector $\mathcal{F} \in \mathbb{R}^n$ is defined as

$$\mathcal{F} = \begin{bmatrix} f_{ext} \\ f_{int} \end{bmatrix} \tag{12}$$

The regular matrix $G \in \mathbb{R}^{n \times n}$ is called the grip matrix, or the grip transform matrix, and is written with W in (1) and N in (8):

$$[G^T]^{-1} = \begin{bmatrix} W \\ --- \\ c_{1,h}^T \\ \vdots \\ c_{n-6,h}^T \end{bmatrix} = \begin{bmatrix} W \\ --- \\ N^T \end{bmatrix} \tag{13}$$

Using this grip transform matrix G, the object external and internal velocities v_{ext} and v_{int} are related to the contact point velocity $d \in \mathbb{R}^n$, which is represented by *Twist Representation* and uses the same coordinates as the wrench representation does [5].

$$d = G^{-1} \mathcal{V} \tag{14}$$

$$\mathcal{V} = G d \tag{15}$$

where

$$\mathcal{V} = \begin{bmatrix} v_{ext} \\ v_{int} \end{bmatrix}, \tag{16}$$

$$v_{ext} = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z] = \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{17}$$

and $v_{int} \in \mathbb{R}^{n-6}$ is the vector of internal velocity deforming the object body. The equations (14) and (15) can be rewritten ([6]) with the definition (13) as

$$d = \begin{bmatrix} W^T & N \end{bmatrix} \mathcal{V} \tag{18}$$

$$\mathcal{V} = \begin{bmatrix} [W^T]^+ \\ N^T \end{bmatrix} d \tag{19}$$

3.2 Force Feedback to Master

In the subsection 3.1, the external force, internal force, external velocity and internal velocity of the grasped object are related to the force and velocity of the finger tips by the grip transform matrix. This subsection shows how to make the force feedback between the slave manipulator described in the subsection 3.1 and a master which has less DOF, one degree of freedom, than the slave has.

The virtual model lies in its own space constructed in a computer and has the scalar position $p_v \in \mathbb{R}$. The authors give a physical dynamics to the virtual model as

$$M_v \ddot{p}_v + D_v \dot{p}_v + K_v p_v = F_m + F_s \tag{20}$$

where M_v, D_v and $K_v \in \mathbb{R}$ correspond to the inertia, damping and stiffness parameter, respectively, about the position p_v, and they can be chosen arbitrary. F_m, $F_s \in \mathbb{R}$ are the generalized forces exerted to the master by the human operator and to the slave by the remote environment, respectively. If p_v means the angular position, F_m and F_s mean the torque.

The authors define the control problem of the master-slave system as described below.

Definition 2 *The position error of master $e_{p,m}(t) \in \mathbb{R}$ is defined as*

$$e_{p,m}(t) = p_m^d(t) - p_m(t) \tag{21}$$

where the scalar $p_m^d \in \mathbb{R}$ is the desired master position and $p_m(t) \in \mathbb{R}$ is the real master position. These positions correspond to the dimension which the human operator wants to control directly with force feeling in the dimensions of the object position.

Definition 3 *The position error of the slave manipulator, $e_{p,s} \in \mathbb{R}^6$, is defined with the position of the grasped object as*

$$e_{p,s}(t) = p_s^d(t) - p_s(t) \tag{22}$$

where $p_s(t) \in \mathbb{R}^6$ is the position of the grasped object, and $p_s^d(t) \in \mathbb{R}^6$ is the desired position.

$$p_s^d(t) = \begin{bmatrix} r_s^d(t) \\ \varphi_s^d(t) \end{bmatrix}, \quad p_s(t) = \begin{bmatrix} r_s(t) \\ \varphi_s(t) \end{bmatrix} \tag{23}$$

$r_s \in \mathbb{R}^3$ designates the vector from the coordinates origin to the center of mass of the object. $\varphi \in \mathbb{R}^3$ represents the roll-pitch-yaw orientation of the object frame of which the coordinates are the principal axis of inertia.

The slave manipulator must exert some internal forces on the object in order to grasp it stably. The necessary internal forces can be calculated from some information: the coefficients of friction of the object surface, the weight of it, etc. We suppose that the magnitude and the orientations of this internal forces are given.

Definition 4 *We define the internal grasp force error of the slave $e_{f,s}$, using the wrench intensity vector c_h included in the subsection 3.1.*

$$e_{f,s}(t) = c_h^d(t) - c_h(t) \tag{24}$$

where the vector c_h^d contains the wrench intensities, which make the desired internal force, in its elements.

Definition 5 *The desired positions given to the master and slave are proportion to the position of the virtual model (20).*

$$p_m^d(t) = k_m p_v(t) \tag{25}$$

$$p_{s,i}^d(t) = \begin{cases} k_s p_v(t) & (i = l) \\ p_{s,i}(0) & (i \neq l) \end{cases} \tag{26}$$

where $p_{s,i}^d \in \mathbb{R}$ and $p_{s,i} \in \mathbb{R}$ means the ith element of the vector p_s^d and p_s, respectively. Only the lth element of p_s^d varies in proportion to p_v, while another element is given an initial value of $p_s(t)$ element. $k_m \in \mathbb{R}$ and $k_s \in \mathbb{R}$ are proportional coefficients about position.

$p_v(t)$ is calculated from the equation (20), while the external forces added on the virtual model, $F_m(t)$ and $F_s(t)$, is measured by the master and the slave system. As the scalar position of the virtual model is given to the only lth element of the desired position vector of the slave, the only lth element of the measured external force vector of the slave f_{ext} is used as the scalar force of the slave in the equation (20):

$$F_s(t) = f_{ext,l} \tag{27}$$

$f_{ext,l} \in \mathbb{R}$ is the lth element of f_{ext}.

Therefore, this virtual model gives the position tracking and the force feedback to the master-slave system about the only one dimension which corresponds to the lth element of p_s and f_{ext}.

Definition 6 (Control Goal) *The goal of the control algorithms in the master-slave system is to assure the position error of master $e_m(t)$, that of slave $e_s(t)$ and the internal grasp force error $e_{f,s}$ to become zero.*

$$e_{p,m}(t \to \infty) \to 0 \tag{28}$$

$$e_{p,s}(t \to \infty) \to \mathbf{0} \tag{29}$$

$$e_{f,s}(t \to \infty) \to \mathbf{0} \tag{30}$$

The master and slave system with the mechanically inherent impedances make their position coincide with the position of the virtual model. The dynamics of the virtual model becomes predominant in the mechanical dynamics of this master-slave system, because the position of the virtual model is determined with the measured forces on the two system by calculation.

3.3 Local Control of Slave and Master

The slave manipulator described in the subsection 3.1 can be controlled with the computed torque method for the grasped object to have the impedance [6],

$$f = K_m \delta\ddot{p} + K_d \delta\dot{p} + K_s \delta p \tag{31}$$

where the matrices K_m, K_d, $K_s \in \mathbb{R}^{6\times6}$ are the impedance inertia, damping and stiffness parameters. $f \in \mathbb{R}^6$ is the resulting generalized force imposed on the center of mass of the grasped object. $\delta p \in \mathbb{R}^6$ is displacement of the position and orientation vector $p \in \mathbb{R}^6$ like the vector defined in (23). We suppose this equation (31) is always asymptotically stable as

$$\delta p(t \to \infty) \to \mathbf{0} \tag{32}$$

when $f = \mathbf{0}$.

The master can be controlled more easily than the slave. When the master has the dynamics as

$$M_m \ddot{p}_m + D_m \dot{p}_m = \tau_m + F_m \tag{33}$$

where M_m, $D_m \in \mathbb{R}$ are the inertia and damping parameter of the master. F_m and $\tau_m \in \mathbb{R}$ are the external force exerted to the master and the force produced by the master. We can control the position of this system to realize the goal (28) by the τ_m,

$$\tau_m = K(p_m^d - p_m). \tag{34}$$

4 Physical Parameter of Virtual Model

As mentioned earlier, the traditional master-slave system can not communicate the force information between the human operator and the remote environment because of the mechanically large impedance of the system. The subsection 3.2 describes that the dynamics of the system can be changed when the system includes the virtual model. If the system uses the virtual model with smaller impedance than the master and slave has, the impedance of the systems becomes smaller than that of the traditional system.

The paper [7] showed that the master-slave system which consists of a virtual internal model, a master and a slave with force sensors became unstable during the slave contacted a stiff environment. Using the one DOF experimental master-slave system, this section shows too small impedance of the virtual model make the system unstable when a load (or an environment) of the slave changes greatly. We can remove this instability with large impedance of virtual model, but this large impedance prevents the light motion during the slave does not has any load.

The subsection 4.1 propose the way to realize a master-slave system that keeps stable when the load of the slave changes and can move lightly during the slave is free. The advantage of this way is shown in subsection 4.2.

4.1 Parameter Determination

The energy stored and lost by the virtual model, $E_v(t) \in \mathbb{R}$, can be split in three parts as

$$E_v(t) = E_M(t) + E_D(t) + E_K(t) \tag{35}$$

where $E_M(t)$, $E_D(t)$, $E_K(t) \in \mathbb{R}$ are the energy stored by the inertia, lost by the damping and stored by the stiffness about the position of this model. They are represented with the position and physical parameters of the virtual model defined in (20):

$$\begin{aligned} E_M(t) &= \frac{1}{2} M_v \dot{p}_v^2(t) \\ E_D(t) &= \int_0^t D_v \dot{p}_v^2(\tau) d\tau \\ E_K(t) &= \frac{1}{2} K_v p_v^2(t) \end{aligned} \tag{36}$$

The human operator and the remote environment put energy into the master-slave system, exerting the force and moving the master and slave. This input energy $E_i(t)$ can be represented with the external forces, $F_m(t)$ and $F_s(t)$, and the positions, $p_m(t)$ and $p_s(t)$, as

$$E_i(t) = \int_0^t F_m(\tau)\dot{p}_m(\tau)d\tau + \int_0^t F_s(\tau)\dot{p}_s(\tau)d\tau \tag{37}$$

$E_v(t)$ must be equal to $E_i(t)$ ideally for all the time t. But these two energy do not coincide, because the positions of the master and the slave differ slightly from that of the virtual model in the real master-slave system.

When this master-slave system become unstable, it produces some power and works against the human operator and the remote environment. All the motion of the system increases $E_v(t)$ by loss of the damping of the virtual model.

The stable motion of the system increase $E_i(t)$: Some part of power from the human operator is lost by the damping, and the remaining power is transmitted from the slave to the remote environment. The equation (37) deals with the power from the system to the remote environment as the negative energy input. Absolute of this energy is smaller than that of the input from the operator.

In the equation (37), unstable motion of the system decreases $E_i(t)$ because the power flows from the system to the human operator and the remote environment.

The authors propose to use these energy information to stabilize the master-slave system as

$$D^* = \frac{\delta E}{\dot{p}_v^2(\tau)\delta\tau} \tag{38}$$

$$D_v = \begin{cases} D^+ & (D^+ < D^*) \\ D^* & (D^- < D^* < D^+) \\ D^- & (D^* < D^-) \end{cases} \tag{39}$$

where $D^* \in \mathbb{R}$ is the new damping parameter of the virtual model and is determined dynamically from $\delta E = E_i(\tau) - E_v(\tau)$ and the velocity $\dot{p}_v(t)$. $\delta\tau$ is a certain small time period on the time τ. D^+ and D^- are an upper and lower limit of D^*.

D^* acts as damper to reduce δE, because D^* becomes large when the system is unstable ($\delta E \gg 0$). Not only D^* makes $\delta E (> 0)$ close to zero, but also $\delta E (< 0)$ close to zero with its negative damping. This negative side of the damping does not relate to the stabilization directly because $\delta E (< 0)$ does not indicate instability, but this negative side is necessary to keep δE nearly equal to zero and to act it as an indicator of the stability with its sign for all time.

4.2 One DOF Slave and Master Experiments

The experimental master-slave system consists of two motors and a controller of the system: the one is master and the other is slave. The two motors are geared DC motors with the torque sensors, and each has the bar attached orthogonally to its drive shaft in order to actuate the external environments (the human operator and the remote environment). Table I shows the mechanical parameters of the master and slave. The positions of the master and the slave are measured with the optical encoders attached to the motor shaft. The sampling time was 0.96[ms].

In this experiments, the slave bar lifted the wired weight (700[g]) as the load which made the torque 0.39 [Nm] by gravity. When the slave put the weight on the table, the slave can move without loads in the remote environment. The human operator changed the load of the slave as he or she varied the position of the master (and the slave).

The Figures 5 and 6 shows the advantage of the parameter determination proposed in the subsection 4.1. The top plots of this two figures show the positions of the master and slave, and the middle show the

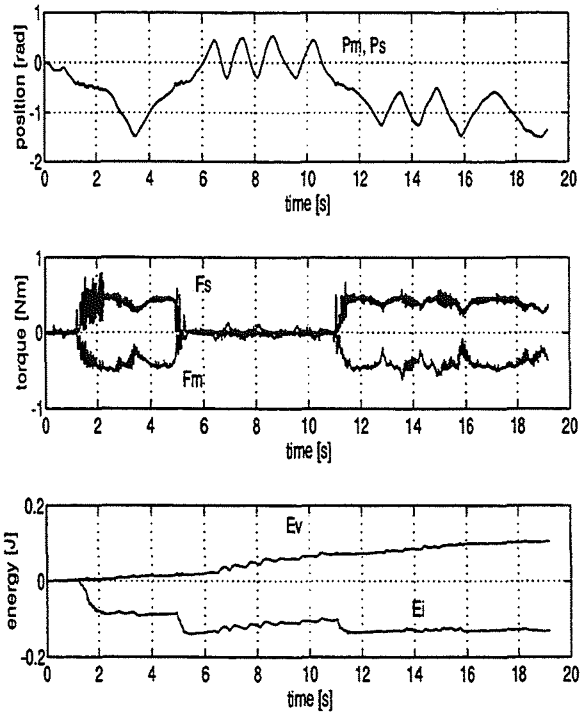


Figure 5: Experimental result with fixed damping parameter of virtual model.

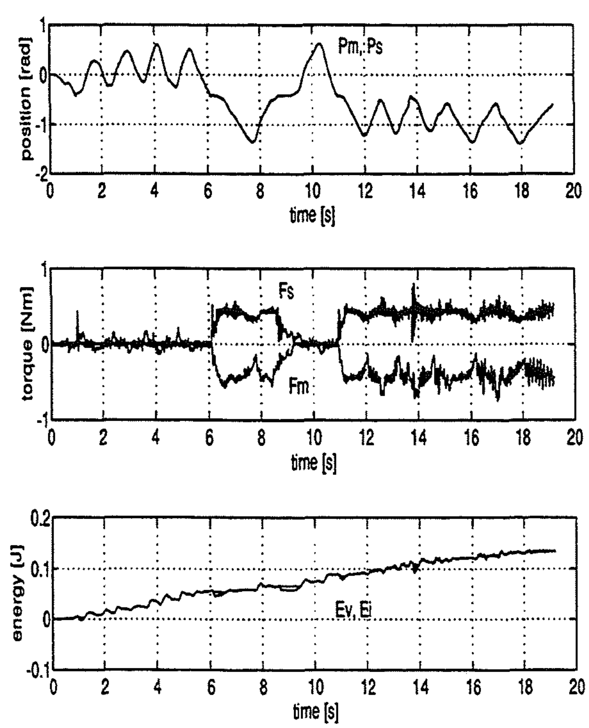


Figure 6: Experimental result with variable damping parameter of virtual model.

Table I: Mechanical parameter of master and slave

	Slave System	Master System
Inertia [kgm^2]	0.020	0.0090
Damping [Nms]	0.44	0.070

Inertia and damping is measured about output side of gear.

Table II: Parameter of Virtual Mode

	Virtual Model
Inertia [kgm^2]	0.005
Damping [Nms]	0.006

torques. The position of the slave coincides with that of the slave, while the torque of the master has the similar form to that of the slave, in each figure.

The considerable difference between the two figures is in the form of the bottom plots. The system showed Figure 5 when it used the fixed damping parameter of the virtual model, as shown in Table II. The inertia and damping of the virtual model were much smaller than that of master and the slave. This small impedance gave the light motion to the system (see the two torques became close to zero during the system moved, the middle plot in the Figure 5). But it also caused oscillation when the load was added on the slave or removed from it. During this oscillation occurred, the input energy to the system, E_i, decreased although the lost and stored energy of the virtual model increased.

The system did not shows this oscillation when it used the dynamically determined damping parameter using the way proposed in subsection 4.1. We set the parameters in (38), (39) as

$$\begin{aligned} \delta\tau &= 0.96\ [ms] : \text{Sampling Time} \\ D^{+} &= 0.3\ [Nms] \\ D^{-} &= -0.3\ [Nms] \end{aligned}$$

The input energy E_i coincided with the energy of the virtual model, E_v, for almost time. E_i became smaller than E_v when the slave was given the load, but E_i coincided with E_v again by the working of the variable damping parameter.

5 Conclusion

The authors include the virtual model which has the physical dynamics, inertia, damping, and stiffness parameter, in the master-slave system. We proposed that the slave system shares the dimensions of its work space with the human operator to make the remote task easy for the human operator. This master-slave system must have the information about the remote task (*e.g.*,the object model in the remote space, the dimensions which the slave can make up for) and must be able to control positions explicitly. We suggest that the one DOF master can control one dimension of the slave work space with the force feedback.

The mechanical dynamics of the master-slave system can be changed with that of the virtual model, which is determined arbitrary. The small impedance of the virtual model leads to the small impedance of the whole system and realizes the accurate communication about the force and position between the human operator and the remote environment. But too small impedance causes instability on the system, and the minimal impedance with stability changes as the slave load changes.

The authors also propose to determine the damping factor of the virtual model dynamically and show the way to do it. This damping factor becomes large to stabilize the system during the system becomes unstable, comparing the input energy to the system with the stored and lost energy of the virtual model. We showed the advantage of the dynamically determined damping by the experiments with one DOF master-slave system.

Acknowledgements

We are grateful to Shinichi Horikoshi for giving us helpful advice about our experimental system. We are also grateful to Dr.Martin Buss for his advice about this manuscript. The computer system in the experimental system was supported by SGS-Thomson Electric Ltd.

References

[1] Goertz, Raymond C., Bevilacqua, Frank, "A Force-Reflecting Positional Servomechanism," NUCLEONICS, Vol.10, No.11, Nov., 1952, pp.43–45.

[2] Goertz, Raymond C., Thompson, W.C., "Electronically Controlled Manipulator," NUCLEONICS, Vol.12, No.11, Nov., 1954, pp.46–47.

[3] Bobgan, Peter M., Kazerooni, H, "Achievable Dynamic Performance in Telerobotic Systems," IEEE International Conference on Robotics and Automation, Sacramento, California., , Apr., 1991, pp.2040-2056.

[4] Sheridan, Thomas B., "Supervisory Control of Anthropomorphic Teleoperators for Space, Undersea and Other Applications," TELEROBOTICS, AUTOMATION AND HUMAN SUPERVISORY CONTROL, The MIT Press, Cambridge, Massachusetts, 1992, pp.99-238.

[5] Salisbury, J.K., Mason, Matthhew T., "Kinematic and Force Analysis of Articulated Hands," ROBOT HANDS AND THE MECHANICS OF MANIPULATION, The MIT Press, 1985, pp.1-167.

[6] Buss, Martin, "Study of Intelligent Cooperative Manipulation," Ph.D.Thesis, University of Tokyo, Sep.,1994.

[7] Furuta, Kastuhisa, Kosuge, Kazuhiro, Shiote, Yoshinori, Hatano, Hiromu, "Control of Master-Slave Manipulator Based on Virtual Internal Model," TRANSACTION OF THE SOCIETY OF INSTRUMENT AND CONTROL ENGINEERS, Japan, Vol.24, No.11, Dec., 1988, pp.176-182.

Physics-based Approach to Haptic Display

J. Michael Brown
J. Edward Colgate

Department of Mechanical Engineering
Northwestern University
2145 Sheridan Rd, Evanston, IL 60208
E-mail : mb@nwu.edu, colgate@nwu.edu

Abstract

This paper addresses the implementation of complex multiple degree of freedom virtual environments for haptic display. We suggest that a physics-based approach to rigid body simulation is appropriate for hand tool simulation, but that currently available simulation techniques are not sufficient to guarantee successful implementation. We discuss the desirable features of a VE simulation, specifically highlighting the importance of stability guarantees.

1. Introduction

A haptic display (or force reflecting interface) is a device which lets the user touch, feel and manipulate virtual environments, rather than just seeing them. As an example, the haptic display of a linear spring must enforce a specific relationship between force and position. Thus if the user grasps the display and applies a certain force, a predictable displacement will result. Many such devices have been developed in recent years, including but not limited to [1, 3, 4, 5, 8, 9, 12, 14, 15, 18, 19].

One promising area for the application of haptic display is tool use, both in terms of the design process and the training of new users. For example, designers could reduce prototyping time and costs by implementing new ideas in a virtual environment, rather than in a machine shop. Conventional VR can be and has been used in this way (see [21] for one example). However, for many tools, appearance doesn't allow a designer to understand how the tool will perform. For this class, functionality is demonstrated by the physical interactions the tool allows between a user and an environment. To explore this functionality, we need the ability to construct and physically interact with virtual environments.

Recently, virtual reality has been used to train Space Shuttle support personnel at Johnson Space Center in procedures that require the use of highly specialized hand tools. While some of these tools are quite ordinary, others have unusual shapes and functions (see Figure 1 for example).

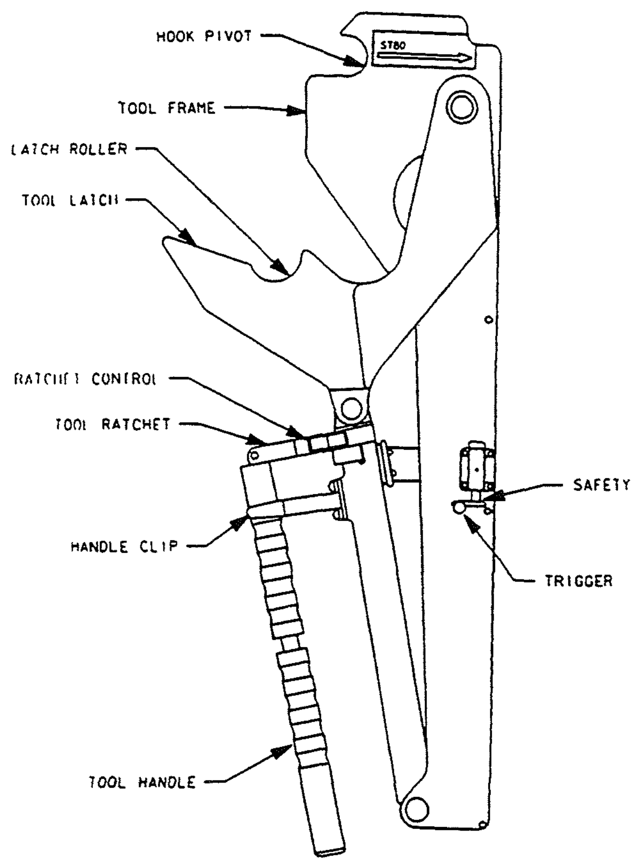


Figure 1. Example of complicated hand tool

However, in the current training environment, tools are not represented at all, since that would require simulation of the interactions between virtual objects. For example, one merely points to a bolt that needs to be loosened, and it loosens itself. Clearly, this is useful for learning a complicated *procedure*, but not a physical *skill* . For simple tools, this is not a problem, but for more complicated ones, the physical skill is a challenging part of the task. To provide astronauts and support personnel with a proper environment for mastering these physical skills, NASA has resorted to using a full-scale mockup of the Shuttle. An alternative to this rather expensive process is to include the hand tools in the VR simulation. Like in the tool design example given above, some tools' functionality cannot be demonstrated with visual information alone. For these tools, haptic interaction is a necessary component of training.

Both of these examples call for an extremely flexible device, capable of being programmed to feel like a wide variety of objects. The flexibility we seek is not just in the device, but in the VE software itself. We would like to be able to adjust parameters quickly and easily, without having to "recompile" the virtual environment.

In a strict sense, the VE software is a real-time simulation of a physical system. It is important that this simulation behave in a physically reasonable manner, because it interacts with two systems (the human user and the handle that he/she grasps) which are, in fact, physical. There are many ways to approach this kind of physics-based simulation, and a vast literature from which to draw knowledge. The next section will review aspects of this literature, specifically addressing the needs of hand tool simulation and haptic display.

2. Rigid body simulation review

Before reviewing simulation techniques, we need to consider the class of physical systems with which we are concerned. We will therefore limit our scope to *rigid body simulation*, which is often appropriate in the context of tool use. However, we need to pay particular attention to each simulation method's ability to deal with *unilateral constraints*, which are ubiquitous in tool use. A unilateral constraint is the type of constraint that typically occurs when two rigid bodies come into contact. It may also be viewed as a bilateral constraint (e.g. a revolute or prismatic joint) that is removed whenever the constraint force becomes negative. Unilateral constraints are challenging to implement because they require a dynamically changing topology (i.e. there is more than one set of motion equations, and which set is enforced depends on the state of the system). With this in mind, there are three major classes of rigid body simulation that we will consider here: constraint stabilization, coordinate partitioning / velocity transformation, and recursive constraint propagation. All three classes assemble a set of motion equations, solve them for accelerations, and integrate to obtain position and velocity.

For constraint stabilization [2, 16, 24], the starting point is the unconstrained equations of motion. Lagrange multipliers are added for each constraint, and the extra equations needed to solve for these multipliers are obtained from the second derivative of the constraint equations. Unfortunately, this technique doesn't precisely enforce the constraints, but rather their second derivatives. Since numerical integration results in finite errors at each time step, the constraint will be violated after just a short period of time. To fix this problem, position and velocity dependent terms can be appended to the constraints' second derivatives, tending to preserve and stabilize the constraint. The difficulty is that picking these position and velocity dependent terms for high accuracy makes the differential equations stiff, mandating smaller time steps to solve accurately. The advantages to constraint stabilization are flexibility and ease of implementation of unilateral constraints. The primary disadvantage is computational cost, so this technique is rarely if ever suitable for simulating complex systems in real time.

Another approach is to identify the constrained degrees of freedom and integrate the remaining equations. The difficulty, of course, lies in identifying those degrees which are constrained and which are free to move. The constraint doesn't even have to be on one of the state variables - it could simply enforce a specific relationship between two of them. A clever approach to this kind of problem is "generalized coordinate partitioning", which automatically extracts the integrable coordinates from the constraints [17, 23]. These coordinates represent non-stiff equations, which can then be integrated easily by any number of techniques. This technique is promising, as it has been used for real-time simulations, and can handle unilateral constraints in a straight forward manner. A drawback of generalized coordinate partitioning is that it expects independent constraints, so the situation shown in Figure 1 could not be allowed without additional logic.

Figure 1. Example of dependent constraints.

Finally, there are recursive techniques, which can provide greater efficiency for certain complicated systems [10, 11, 13, 22]. However, they require topological preprocessing, meaning the connectivity of the bodies must be assessed and a computational hierarchy established beforehand. This type of preprocessing eliminates the possibility of a dynamically changing topology, so extra provisions must be made to govern collisions between bodies.

While one of these approaches to rigid body simulation may provide a suitable starting point for a VE hand tool simulator, it must be appreciated that haptic display introduces certain additional considerations. Specifically, haptic display differs in three key ways : real-time processing (as already mentioned), high update rates and stability guarantees.

Due to its interactive nature, haptic display requires real-time processing, a problem it shares with conventional virtual reality. Conventional VR, however, is not typically physics-based, so this requirement, while posing a problem in terms of video update, isn't too difficult for the simulation itself to handle. For physics-based simulation, the need for real-time processing eliminates constraint stabilization as a likely choice, since it requires solution of stiff differential equations, a difficult process in real time.

Haptic display has the additional requirement of high update rates because of the bandwidth of human tactile senses (upwards of 1 KHz). This problem is not shared by conventional VR, since the bandwidth of human optical senses is around 70 Hz. This is not to say that visual VR is easier than haptic VR, it just has a different set of challenges to overcome.

A primary goal of VR, whether haptic or visual, is to try to achieve "presence" in a virtual environment [20]. If the state of that environment becomes computationally unstable, the sense of presence will be damaged, if not completely destroyed (imagine if a wrench began oscillating uncontrollably against a nut). Thus, physics based VR, whether haptic or visual, needs to provide a stability guarantee. None of the methods described above can provide this guarantee. Our experience has shown that for haptic display, stability guarantees are the most challenging aspect of virtual environment implementation.

Our long-term goal is to design a haptics programming language which allows complex VEs to be rapidly assembled and modified, while providing stable, realistic interaction. Since all three of the above approaches have problems, we have begun investigating techniques which utilize parallel processing to achieve this goal. In the remainder of this paper, however, we discuss the problem of providing stability guarantees, rapidly becoming recognized as the sine qua non of haptic display.

3. Providing a stability guarantee

We believe two components are necessary for the haptic display of complex multiple degree of freedom tool simulations :

- The ability to display a set of haptic primitives. This set includes, but is not limited to, springs, viscous drag, inertia, friction and hard non-linearities.
- The ability to connect these primitives arbitrarily and still guarantee stability of interaction. Particularly important is the ability to implement unilateral and bilateral constraints.

Complex environments can be broken down into smaller simpler components called "primitives". A haptic primitive may be described as a mechanical impedance, a relationship, possibly history-dependent, between motion and force. Unfortunately, reliable display of such a primitive involves issues of safety as well as accuracy of display. Because the user, manipulandum, actuators and virtual environment form a dynamic system, stability of this system becomes an issue. We need an intellectual framework to predict the behavior of this system.

To ensure robust interactive behavior, as in the example of the wrench and the bolt, the physical world relies heavily upon the property of *passivity*. The wrench and bolt are obvious examples of passive systems, neither being able to provide energy to the other. It is well-known that the coupling of passive systems is guaranteed to be stable. Furthermore, humans are adept at manipulating passive objects in a safe and efficient manner. In our studies of virtual walls, we have found that passivity provides an extremely useful intellectual framework for understanding the stability problem.

In order to investigate the passivity of haptic virtual environments more closely, we built a one degree of freedom manipulandum [6], shown in Figure 2.

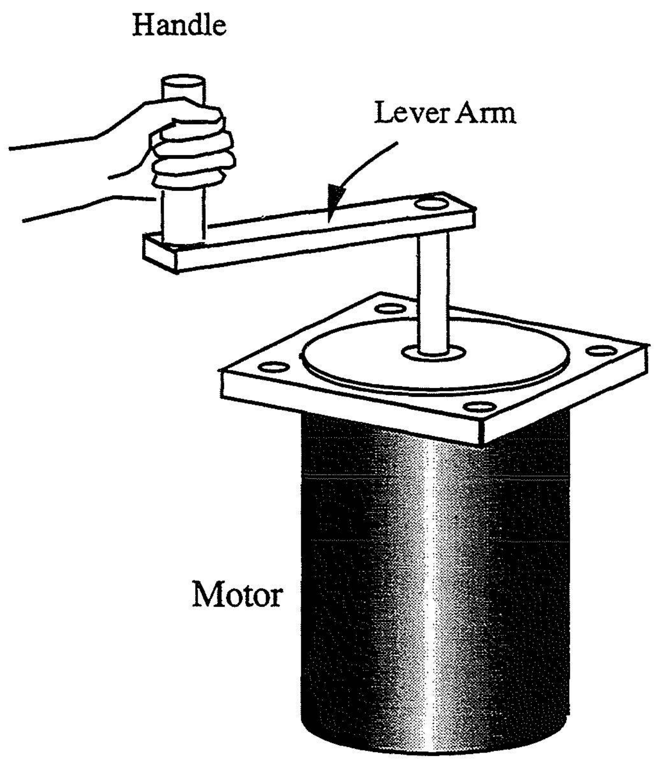


Figure 2. 1 DOF haptic display

The manipulandum is powered by a PWM-driven DC brushless motor. Position sensing is provided by optical encoders on the motor shaft. Controlled only by a 486 50-MHz PC, the system is capable of updating simple haptic virtual environments at up to 10 KHz. Graphic representation of the virtual environment is displayed on a 15-inch color monitor. A model of the system dynamics is shown in Figure 3.

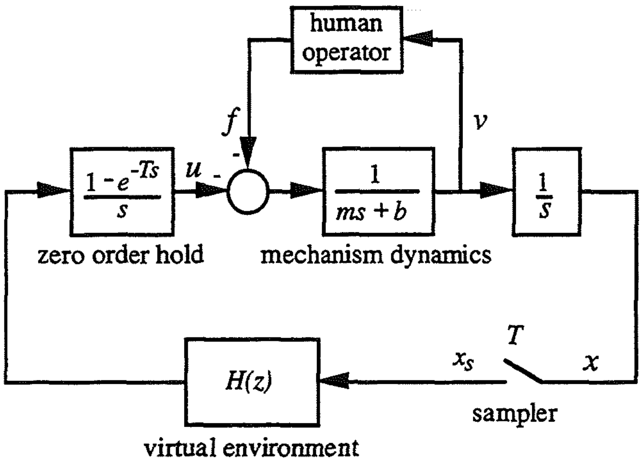


Figure 3. Model of a one degree-of-freedom haptic interface. m is the inherent mass of the display, b is inherent damping, H(z) is the virtual environment transfer function, v is velocity, x is position, x_s is the sampled position, T is the sampling rate, u is the control effort, and f is the force applied by the operator.

The difficulty in any traditional stability analysis of this system is the unmodeled dynamics of the human operator. Even though the virtual environment itself might be stable, interaction with a human operator via a haptic interface may *cause* instability. In our studies of virtual environments, we have had many experiences with human operators adjusting their own behavior until oscillations resulted. However, if the display is truly passive, then human operators should not be able to destabilize the system. If this approach is taken with the model presented in Figure 3, the following theorem, proven in [7], results :

Theorem — A necessary and sufficient condition for passivity of the haptic interface model in Figure 2 is:

$$b > \frac{T}{2}\,\frac{1}{1-\cos\omega T}\,\mathrm{Re}\left\{(1-e^{-j\omega T})\,H(e^{j\omega T})\right\} \qquad \text{for } 0 \le \omega \le \omega_N \qquad (1)$$

Here, *b* is the inherent damping of the display, *T* is the sampling rate, *H(z)* a pulse transfer function representing the virtual environment, and $\omega_N = \pi/T$. This theorem shows that inherent physical damping is required to make a haptic display passive. This result goes against the conventional wisdom of haptic display design that a device have minimal inherent friction and damping.

Virtual Walls

It is important to note that a haptic display may be called upon to exhibit a wide variety of impedances, including those which are highly nonlinear. As a specific but enlightening example, we consider the virtual wall. The virtual wall can be modeled with three haptic primitives, a stiff spring, a damper and a hard non-linearity, implemented in parallel (see Figure 4).

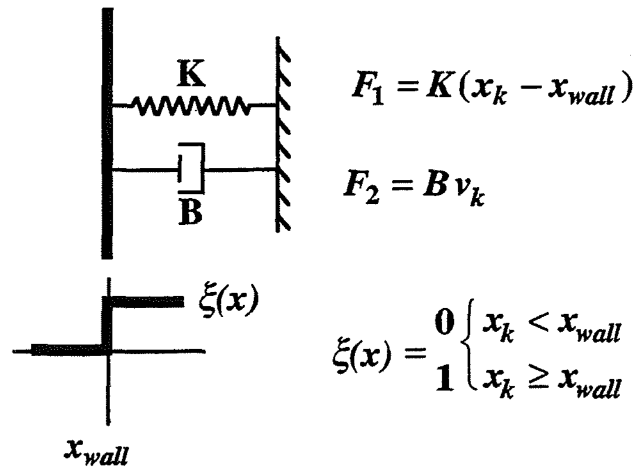


Figure 4. Model of a virtual wall as a spring and damper in parallel. K defines the virtual stiffness, B the virtual damping, and x_{wall} the location of the wall.

such that the total force experienced by the operator is given by :

$$F = \xi(F_1 + F_2) \qquad (2)$$

The virtual wall is extremely challenging to implement since it includes the extremes of impedance, along with rapid transitions between them. Outside the wall, the operator should be able to move the device freely (low impedance), but inside the wall, the operator should be unilaterally constrained (high impedance). The device needs to be able to implement both of these extremes and be able to switch between them almost instantaneously. We feel that if a system can successfully simulate contact with hard surfaces, it possesses the *dynamic range* to display the results of many useful virtual environments. Substituting the specific equations for a virtual wall, (1) reduces to :

$$b > \frac{KT}{2} + |B| \qquad (3)$$

where b is the inherent physical damping of the device, K and B are the virtual stiffness and damping, and T is the sampling rate. Based on (3), it is easy to see that inherent damping and sampling rate have significant effects on the passivity of virtual walls. Another more subtle factor that doesn't show up in this analysis is the effect of sensor resolution on system performance. The effect of these factors on stability was quantitatively assessed in [6]. The results are summarized as follows :

- Inherent physical damping of the haptic display improves passivity
- High update rates increase achievable stiffnesses of virtual walls
- If encoders are used to estimate velocity, they should have extremely high resolution
- Digital filtering of the velocity signal can help achieve high values of virtual damping

4. Conclusions

Based on these guidelines (obtained with a 1 DOF device), we have equipped a 4 DOF manipulandum [14] with dampers, allowing us to construct multi-DOF virtual environments with convincing unilateral constraints. However, no method of guaranteeing system stability has yet been found. As mentioned above, our current research is focused on the development of a haptics programming language which will utilize parallel processing. This language will allow complex VEs to be rapidly assembled and modified, while providing stable, realistic interaction with the human operator.

5. Acknowledgments

The authors gratefully acknowledge the support of the National Science Foundation, grants MSS-9022513 and IRI-9213234 and NASA, grants ..., as well as valuable discussions with Mike Stanley, Witaya Wannasuphoprasit, Paul Millman, Beeling Chang and Jui-Chang Tsai.

6. References

1. Adelstein, B. D. and M. J. Rosen. *Design and Implementation of a Force Reflecting Manipulandum for Manual Control Research.* ASME Winter Annual Meeting. Anaheim, California (1-12) (1992)

2. Barzel, R. and A. Barr. *A Modeling System Based on Dynamic Constraints.* Computer Graphics **22**(4):179-187 (1988)

3. Bergamasco, M. *Theoretical Study and Experiments on internal and External Force Replication. IEEE Workshop on Force Display in Virtual Environments and its Application to Robotic Teleoperation.* Atlanta, Georgia (1993)

4. Brooks, F. and e. al. *Haptic Displays for Scientific Visualization.* Computer Graphics **24**(4):177-185 (1990)

5. Burdea, G., J. Zhuang, E. Roskos, D. Silver and N. Langrana. *A Portable Dextrous Master with Force Feedback.* Presence **1**(1):18-28 (1992)

6. Colgate, J. E. and J. M. Brown. *Factors Affecting the Z-width of a Haptic Display.* International Conference on Robotics and Automation. San Diego, CA (3205-10) IEEE R&A Society (1994)

7. Colgate, J. E. and G. G. Schenkel. *Passivity of a Class of Sampled-Data Systems: Application to Haptic Interfaces.* American Control Conference. Baltimore (1994)

8. Ellis, R. E., O. M. Ismaeil and M. G. Lipsett. *Design and Evaluation of a High-Performance Prototype Force-Feedback Motion Controller.* Advances in Robotics, Mechatronics and Haptic Interfaces, 1993. Kazerooni, Colgate and Adelstein ed. ASME. (1993)

9. Fasse, E. D. and N. Hogan. *Quantitative Assessment of Human Perception of Virtual Objects.* Advances in Robotics, Mechatronics and Haptic Interfaces, 1993. Kazerooni, Colgate and Adelstein ed. ASME. (1993)

10. Featherstone, R. *The Calculation of Robot Dynamics Using Articulated-Body Inertias.* The International Journal of Robotics Research **2**(1):13-30 (1983)

11. Hollerbach, J. M. *A Recursive Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity.* IEEE Trans. on Systems, Man, and Cybernetics **SMC-10**(11):730-736 (1980)

12. Iwata, H. *Artificial Reality with Force-feedback: Development of Desktop Virtual Space with Compact Master Manipulator.* Computer Graphics **24**(4):165-170 (1990)

13. Luh, J. Y. S., M. W. Walker and R. P. C. Paul. *On-Line Computational Schemes for Mechanical Manipulators.* ASME Journal of Dynamic Systems, Measurement and Control **102**:69-76 (1980)

14. Millman, P. A. and J. E. Colgate. *Design of a Four Degree-of-Freedom Force-Reflecting Manipulandum with a Specified Force/Torque Workspace.* IEEE International Conference on Robotics and Automation. Sacramento, CA (1488-1493) (1991)

15. Minsky, M., M. Ouh-young, O. Steele, J. F.P. Brooks and M. Behensky. *Feeling and Seeing: Issues in Force Display.* Computer Graphics **24**(2):235-243 (1990)

16. Nikravesh, P. E. *Some Methods for Dynamic Analysis of Constrained Mechanical Systems: A Survey.* Computer Aided Analysis and Optimization of Mechanical System Dynamics. Haug ed. Springer-Verlag. New York (1984)

17. Park, T. W. and E. J. Haug. *A Hybrid Numerical Integration Method for Machine Dynamic Simulation.* Journal of Mechanisms, Transmissions, and Automation in Design **108**:211-216 (1986)

18. Rosenberg, L. B. and B. D. Adelstein. *Perceptual Decomposition of Virtual Haptic Surfaces.* IEEE Symposium on Research Frontiers in Virtual Reality. San Jose, CA (1993)

19. Salcudean, S. and N. M. Wong. *A Force-Reflecting Teleoperation System with Magnetically Levitated Master and Wrist.* Proc. IEEE ICRA. Nice, France (1420-1426) (1992)

20. Slater, M. and M. Usoh. *Presence in Immersive Virtual Environments.* IEEE Virtual Reality Annual International Symposium. Seattle, Washington (90-96) (1993)

21. Tanner, S. *The Use of Virtual Reality at Boeing's Huntsville Laboratories.* IEEE Virtual Reality Annual International Symposium. Seattle, Washington (14-19) (1993)

22. Walker, M. W. and D. E. Orin. *Efficient Dynamic Computer Simulation of Robotic Mechanisms.* Journal of Dynamic Systems, Measurement and Control **104**:205-211 (1982)

23. Wehage, R. and E. J. Haug. *Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems.* Journal of Mechanical Design **104**:247-255 (1982)

24. Witkin, A., M. Gleicher and W. Welch. *Interactive Dynamics.* Computer Graphics **24**(2)(1990)

THERMAL FEEDBACK IN VIRTUAL REALITY AND TELEROBOTIC SYSTEMS

Mike Zerkus, Bill Becker, Jon Ward, Lars Halvorsen

RESEARCH
2437 Bay Area Blvd. #234
Houston, TX 77058
1-800-262-1CMR 713-488-3598 713-488-3599 (FAX)

ABSTRACT

A new concept has been developed that allows temperature to be part of the Virtual World. The Displaced Temperature Sensing System (DTSS) can "display" temperature in a virtual reality system. The DTSS can also serve as a feedback device for telerobotics.

For Virtual Reality applications the virtual world software would be required to have a temperature map of its world. By whatever means (magnetic tracker, ultrasound tracker, etc.) the hand and fingers, which have been instrumented with thermodes, would be tracked. The temperature associated with the current position would be transmitted to the DTSS via a serial data link. The DTSS would provide that temperature to the fingers[1].

For Telerobotic operation the function of the DTSS is to transmit a temperature from a remote location to the fingers where the temperature can be felt.

DISPLAY THEORY

By simply thinking of the many languages and forms of writing one comes to the inescapable conclusion that there are many way to present the same idea. The clarity of that presentation is a function of the individual; weather or not he knows the language, his background and so on. The display of a machine, be it as simple as the calibrated marks surrounding the volume knob of a radio or a Heads Up Display (HUD) of a fighter plane, is supposed to translate information into a form we can understand. Thus, *clarity* is still a function of the individual; weather or not he knows the language, his background (training), etc.

Information has 2 basic types, *inherent* and *abstract.* Inherent information is information that is common to all humans. For example, hot, cold, loud, rough, smooth are common to all humans, regardless of how they may be expressed. Abstract information is text, graphics and other things that require interpretation and prior knowledge.

---

[1]Obviously other parts of the body could be fitted with thermodes, but we don't talk about that.

From the above 4 laws are realized.

1) The usefulness of a machine is determined by the ability of the display of that machine to convey information within that machine.
2) Any display can do the job of any other display.
3) The ability of a display to reproduce the actual situation makes that display useful.
4) The perception of abstract information presented by a display is culturally determined.

PRESENTING TEMPERATURE INFORMATION.

The notion of temperature is implied in our language that describes reality: a summer day, a winter storm, a cup of coffee, or a drink at the water fountain. Thermal sensation gives other cues to the nature of things in the environment around us; for example, the average person can easily tell the difference between metal and wood because the difference in the thermal conductivitys is felt as apparent cold. Temperature is inherent information and therefore best displayed as hot and cold .i.e. felt as hot and cold. Reality is not complete without temperature. It fills in our picture of reality with the details that make everything seem correct.

The DTSS allows use of physiology deception to enhance realism of the virtual world. In addition to presenting thermal information **Weber's Deception** can be used to create the sensation of touching an object. Weber's Deception is the sensation of pressure or contact caused by slightly cooling the skin.[6]

THERMOELECTRIC HEAT PUMPS

A thermoelectric heat pump (sometimes called a Peltier Device or a thermoelectric cooler) is a solid state device that moves heat from its cold side to its hot side. Thermoelectric heat pumps are heat pumps just like the mechanical heat pumps used in refrigeration or air conditioning except they have no moving parts. All of the thermodynamic laws that govern conventional heat pumps also govern thermoelectric heat pumps.

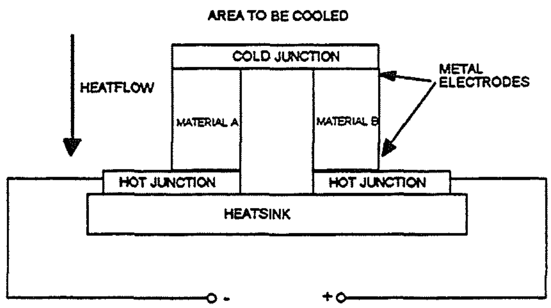


Figure 1.
A single element thermoelectric heat pump.

A thermoelectric heat pump can be thought of as a thermocouple being driven backwards. A thermocouple is a common temperature measurement device consisting of a junction of 2 dissimilar metals. When the junction is heated an electric current is produced. In a thermoelectric heat pump there are 2 junctions (see Figure 1). One junction is located in or on the space to be cooled; the other junction is located on the heat sink. When voltage is applied, the temperature of the junction in the space to be cooled will decrease and the temperature of the other junction will increase and heat will be transferred from one side to the other. The thermoelectric process is reversible. If the current through the heat pump is reversed the cold side becomes the hot side and heat flows in the opposite direction. A typical thermoelectric heat pump can generate up to a 67°C temperature difference from on side of the heat pump to the other[2]. Heat pumped is roughly proportional to the current through the heat pump[3].

THERMODES

A thermode is an assembly consisting of a thermoelectric heat pump, a temperature sensor, and a heat sink. The heat pump moves heat into or out of the heat sink to produce a temperature at the surface of the thermode. Using feedback from the sensor, the DTSS regulates the temperature of the surface of the thermode. A thermode can also serve as an input, sensing temperature and surface thermal conductivity.

The basic physical configuration of a thermode is shown in figure 2. A temperature sensor is mounted on top of the thermoelectric heat pump. The temperature sensor provides feedback to the control network. The heat sink is in contact with ambient temperature air.

---

[2]The 67°C temperature difference is for a no load condition.

[3]This is not strictly true; thermoelectric heat pumps are not linear, but do have regions where they are near linear. There is also a performance difference between heating and cooling.

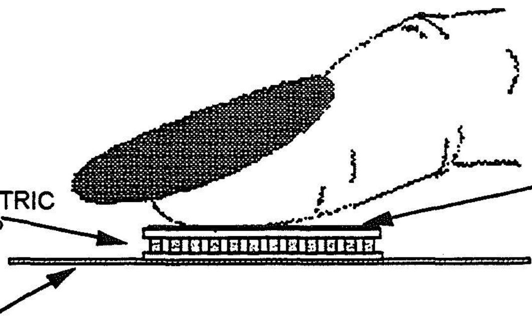


Figure 2.
Basic physical configuration of DTSS Thermode.

SAFETY

Thermoelectric heat pumps in contact with human skin can cause burns. Any experimentation with thermoelectric heat pumps should provide a way of preventing burns.

The comfort zone for humans is from 13°C to 46°C, with pain below and above these limits. The average human can feel a temperature change as little as 0.1°C over the entire body, however, at the finger tip a sensitivity of 1°C is typical. Exact numbers vary from person to person. [7]

A Thermal Electric Heat Pump used to stimulate thermai sensation to fingertips has several inherent safety problems.

The finger contains heat which must be dissipated in order for a person to feel cool. Because heat is convected to air (through the heat sink) slower than heat conducted from the finger, the heat sink size for the thermoelectric heat pump has to be large enough and have enough surface area so that the heat sink is not overwhelmed. If the heat sink is overcome (usually because the heat pump was operated in cooling mode for an extended period of time), the heat pump can not maintain the temperature difference. The heat in the heat sink will come back through the heat pump and burn the finger.

Another potential safety problem occurs if the heat pump is operated in a cooling mode for an extended period of time and the power to the unit fails. In such a situation the unpowered heat pump becomes a sandwich of ceramic and metal (with good heat conductivity), and once again, the heat in the heat sink flows back through the heat pump and burns the finger.

CONTROL SYSTEM

Figure 3 shows a block diagram of the control system used for the DTSS. The goal of the control system is to have the temperature at the finger tip follow the temperature command.

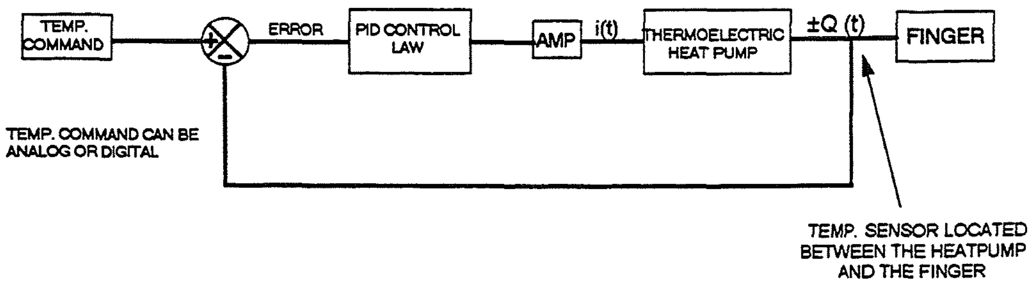


Figure 3.
Block diagram of the DTSS control system.

An early DTSS prototype used a proportional control law. It was found that in order to have an effective response time the gain had to be very high, but this caused temperature ringing at the fingertip (a very weird physical sensation). The DTSS uses a **Proportional Integral Derivative** (**PID**) control law for closed loop control of thermode temperature. A PID control law allows gross temperature error, cumulative error and oscillation to all be controlled. The control law is implemented in the software.

FEATURES

CM Research's first DTSS product is the model X/10. The X/10 is designed as a research unit for those who want to add temperature to their work.

- The X/10 has eight thermode channels. Each channel is software programmable as an input or an output. The inputs can be "mapped" to outputs, such that the output temperature tracks the input temperature; this is called analog track mode. Any input can be mapped to any output or group of outputs.
- The DTSS can be operated from the front panel or remotely via an RS-232. A front panel is provided so the unit can be used in a stand alone configuration. The front panel also makes troubleshooting easier in situations where the X/10 is part of a larger system.
- Differential analog inputs are provided so the X/10 can track an analog signal from some external device.
- The gains of each part of the PID control law (P, I, and D) are software adjustable via the front panel or the serial communications port.
- Demonstration software (with source code) is included to provide examples for interfacing the X/10 to other systems.

DTSS X/10 Safety Features
CM Research has elected to develop an intrinsically safe thermode. The temperature of the heat sink is never allowed to go beyond 45°C. This results in some degraded long term performance, but provides a simple way to overcome the safety problems mentioned above.

- The DTSS X/10 temperature reproduction range is 10°C to 45°C, with an ambient temperature operating range from 10°C to 35°C. By operating within the comfort zone for humans, the temperature differences are kept small, which allows for better use of energy.
- The size of the heat sinks are designed for maximum surface area.
- Power to the thermode has to be actively engaged by the computer after computer power up.
- Thermostats on the thermode cut power to the thermode if the heat sink or the surface of the thermode exceed 45°C.
- Redundant safety software zeros the input to the thermode if the operating range is exceeded.

APPLICATIONS
In a telerobotics application, temperature sensors could be placed in the fingers of remote manipulators. Temperature signals would be sent to the DTSS and drive thermodes on the fingers of the operator. The DTSS X/10 can accept analog input as well as serial digital input.

A virtual reality application would not require a temperature sensor input; the DTSS would take serial digital commands from the computer controlling the simulation. For example, thermodes would be placed on the fingers of the virtual explorer and a temperature value would be assigned to objects or locations in the virtual world. As the hand moved near these objects, commands would be sent via digital serial communications to the DTSS to change the temperature of the thermodes.

Prosthetics research applications; the DTSS X/10 can be used by researchers to explore application of displaced sensing to prosthetics. Temperature sensors could be placed in the fingers of the prosthetic limb and the displaced sensing system would be used to transmit the temperature felt by the prosthetic fingers to some point on the body, where the temperature could be felt.

CONCLUSION.

Another building block for the virtual world has been developed, thus another aspect of reality can be simulated.

ACKNOWLEDGMENTS

We would like to acknowledge the help and contributions of Randy Martin, Liz Eichaelman, Nan Crowhurst , Kathey Haynes, Don Barr and Stormchylde Borsetti.

REFERENCES

1. *Application Notes for Thermoelectric Devices*, Melcor Corp., Trenton NJ., 1985
2. J. H. Seely, *Elements of Thermal Technology*, Marcel Dekker, Inc., 1981.
3. M. Kutz, *Temperature Control*, John Wiley & Sons 1968.
4. J. J. Distefano III, A. R. Stubberud and I. J. Williams., *Feedback and Control Systems*, McGraw - Hill, 1967.
5. T. R. McKnight, The effects of sinusoidal ripple current upon the temperature difference across a thermoelectric cooling device, a report prepared by the U. S. Naval Ordinance Laboratory, White Oak, Maryland, March 1965.
6. C. T. Morgan, Physiological Psychology, McGraw-Hill Inc., New York, 1965.
7. Guyton, A. C., Text Book of Medical Physiology, W. B. Saunders Company, Philadelphia.

VIRTUAL ENVIRONMENT APPLICATION WITH PARTIAL GRAVITY SIMULATION

David M. Ray & Michael N. Van Chau
NASA - Lyndon B. Johnson Space Center
Space and Life Sciences Directorate
Mockup and Trainer Group
Mail Code: SP52
Houston, TX 77598

ABSTRACT

To support future manned missions to the surface of the moon and Mars and missions requiring manipulation of payloads and locomotion in space, a training facility is required to simulate the conditions of both partial and microgravity as compared to the gravity on Earth. A partial gravity simulator (Pogo), which uses pneumatic suspension, is being studied for use in virtual reality training. The Pogo maintains a constant partial gravity simulation with a variation of simulated body force between 2.2% and 10%, depending on the type of locomotion inputs. This paper is based on the concept and application of a virtual environment system with the Pogo.

The virtual environment system includes a head-mounted display and glove. The reality engine consists of a high-end SGI workstation and PC's which drive the Pogo's sensors and data acquisition hardware used for tracking and control. The tracking system discussed is a hybrid of magnetic and optical trackers which are being integrated for this application. Future upgrades are planned for the facility to further increase the sense of immersion it provides to the subjects training in the virtual environment.

INTRODUCTION

Virtual Reality (VR), or Virtual Environment (VE), systems have come a long way in the past several years. Initially, users could only immerse themselves visually in an environment where their head movements were tracked mechanically. Today there are a variety of tracking options, besides mechanical methods, which do not encumber the user giving them greater freedom with natural motion. In addition, gloves have become available which allow users to interact with virtual objects in their environment. Many researchers are working to include sensory feedback, such as temperature and pressure, through these gloves. Furthermore, three-dimensional audio systems are now available which allow users to hear spatialized sounds in their environment through headphones increasing the level of immersion the user has in the virtual environment. Researchers and end-users now have audio, visual, and some sensory feedback in their interactive virtual environments.

At NASA's Lyndon B. Johnson Space Center (JSC), research is being conducted towards increasing the level of sensory immersion in a virtual environment by merging present VR hardware capabilities with a partial gravity simulator. This application would allow users to interact within a virtual environment while physically experiencing microgravity to some degree. The purpose for providing partial gravity simulation on Earth is for crew safety. The more experience an astronaut has with microgravity as he or she prepares for a mission, the less the chances of risk or mishap that will occur during the actual mission.

Partial gravity simulation techniques will be described to some degree with greater detail provided on the Pogo partial gravity simulator. The concepts discussed will provide a better understanding of the significance and timeliness of the VE application. A description of the VE architecture that is being developed for the Pogo will follow with some mention of future plans underway. The conclusion will state the findings and status of the research that has been completed as of the submission of this article (October 1994).

PARTIAL GRAVITY SIMULATION TECHNIQUES

At present, NASA has two frequently utilized techniques for providing astronauts with a physical experience of microgravity. The first is through the KC-135. The KC-135 is a modified aircraft which allows a person to experience weightlessness or microgravity by flying parabolic trajectories. The second method is by using the Weightless Environment Training Facility, where suited astronauts are made neutrally buoyant underwater by attaching ballasts on their suits. These two methods serve their purpose for specific training applications, but they each have their advantages and disadvantages.

KC-135 Aircraft

The KC-135 has the advantage of providing true zero-g or partial-g simulation. As the aircraft reaches its apogee and begins to descend, the crew can experience a wide range of microgravity depending on the slope of descent. In addition, the aircraft provides a comfortable shirt sleeve environment for studying the effects of microgravity on the crew.

Unfortunately, the simulation is limited by the amount of time an astronaut can execute a particular action. If zero-g conditions are desired, the total duration of simulation is approximately 30 seconds, of which only 20 seconds is available for useful data. If Lunar gravity, or 1/6-g is desired, 30 seconds is the period available to take data. If Martian gravity, or 3/8-g is desired, the useful period is about 40 seconds. The disadvantages due to the short duration of the simulation is obvious. In addition, there are restrictions due to the internal volume of the aircraft. A crew member is limited to a volume 79 in. high, 36 in. wide, and 247 in. long. Finally, the parabolic trajectory, which is executed repeatedly on a flight, can easily induce motion sickness. The KC-135 happens to be nicknamed the "Vomit Comet."

Weightless Environment Training Facility

The Weightless Environment Training Facility , or WETF, simulates partial gravity using Archimedes principle applied to water buoyancy. Its advantages are the full range of degrees-of-freedom it offers in a fairly large volume without the need for mechanical support structures. The disadvantages include the resistance to body motion due to hydrodynamic drag and the limitations on training hardware that can be used due to the corrosive effects of the conditioned water. In addition, simulating lunar or planetary surfaces is very difficult and not practical in a water facility. Finally, the crew has the same dangers that divers must face anytime they remain submerged in the water.

Besides the above methods of microgravity simulation, there have been a number of suspension techniques (which will not be described here in detail) including inclined plane suspension, counterbalance suspension, bungee cord suspension, and pneumatic suspension. The partial gravity simulator which is being used for this research is called the Pogo, which uses pneumatic suspension.

THE POGO PARTIAL GRAVITY SIMULATOR

The Pogo system is a combination of hardware salvaged from a partial gravity simulator used during the Apollo program and state-of-the-art data acquisition and control equipment added during current system development and testing efforts. Pogo consists of three major systems: (1) the vertical servo system; (2) the display and control system; and (3) the gimbal support system. The vertical servo system, shown in Figure 1, provides control of the pneumatic actuator by using servovalve amplifiers. The vertical servo system and the gimbal support system and their principles of operation will be described before presenting the research activities proposed for the virtual environment system integration with the Pogo.

Vertical Servo System Description

The vertical servo system is the mechanism which applies a constant lifting force opposite in direction to the Earth's gravity vector. The vertical servo system consists of: (1) the vertical servo assembly; (2) the cylinder assembly; and (3) the piston rod assembly. Lifting force is provided by supplying the cylinder with pressurized air regulated by the vertical servo assembly. The available air supply to the vertical servo assembly has a maximum pressure of 120 psig (lbs. per sq. in. gage) or 134.7 psia (lbs. per sq. in. absolute) and a maximum flow rate of 367 scfm (standard cubic feet per minute).

Gimbal Support System

The gimbal support assembly is the structure in which training participants are placed to provide the rotational degrees-of-freedom of pitch, roll and yaw. The gimbal support assembly is constructed of aluminum for the structural members and either nylon or kevlar webbing for the support straps. Kevlar is used due to its excellent strength-to-weight ratio and its high resistance to deflection or stretching. Minimal deflection is important because forces stored in the straps, due to elastic properties while deflecting under loaded conditions, will adversely affect the partial gravity simulation. Once a training participant is placed on the seat support and strapped into the chest harness, adjustments are made to insure the body center of gravity coincides with the pitch, roll and yaw axes of the gimbal. A full 360° rotational freedom is capable about the pitch and yaw axes, but the rotation about the roll axis is limited to +/-30°.

Vertical Servo Flow System Description

Maintaining stability in pneumatic systems is a problem when designing closed loop pressure and flow controls. Harmonic oscillations or whistles can be generated, given certain flow conditions coupled with changing line diameters, nozzles and orifice restrictions when compressed air is transmitted. Such conditions are prevalent in the vertical servo flow control system, which is shown schematically in Figure 2.

According to Burrows [3], "The main goal in designing a control system is to achieve adequate dynamic performance without the system becoming unstable." One of the design goals in developing a stable control system for the vertical servo is to determine the best combinations of supply pressure and flapper-nozzle control valve gap settings that result in stable performance of the two-stage mechanical amplification feedback of the Pogo vertical servo. The Pogo vertical servo is a pressure and flow regulating device in which the amplification is error actuated. To operate properly, the vertical servo needs to be a fast responding regulator, where the desired lift force from the piston/cylinder lifting actuator is maintained constant for a continuously varying input load at the end of the actuator.

The first step in developing a stable operating control system is to define the system. A control block diagram of the vertical servo flow control system is shown in Figure 3. The first stage of the vertical servo consists of the flapper-nozzle control valve, and the second stage consists of the intake and exhaust servovalves. An instantaneous change in pressure (P_{ao}) in the cylinder, due to training participant motion, is compared to the desired input lifting pressure (P_{ai}) for constant partial gravity simulation. The result of this comparison is the error signal (ε). The error is amplified by the flapper-nozzle control valve element (F_b), which represents the influence of the bias spring force of the vertical servo. The flapper-nozzle controller in turn affects the back pressure (P_b) in the intake and exhaust servovalves. The back pressure (P_b) is considerably less than the control pressure (P_c), due to the orifice restriction at the inlet to each servovalve control chamber. The servovalve amplifier acts as a second stage pressure regulating element, which further amplifies the error signal and supplies the required pressure change to the lifting cylinder to reduce the difference between (P_{ai}) and (P_{ao}).

The vertical servo acts basically as a two-way flow and pressure regulator. The main supply of air flow enters the intake of the valve chamber of the intake servovalve and is then diverted to two different directions. One direction is toward the inlet to the lifting cylinder and the other is toward the inlet to the exhaust servovalve. The amount of air flow going to either the cylinder or to the exhaust is proportional to the back pressure (P_b), which varies according to the position of the flapper between the control nozzles. The error signal (ε) is directly proportional to the position of the flapper between the control nozzles.

VE Applications with Pogo

The advantages of using a virtual environment with the Pogo is that visual and audio cues can be coupled with full body motion. This effect obviously increases the sense of immersion within the environment. In addition, one can easily change the virtual environment by loading the needed environment database into the reality engine allowing for various training scenarios to be exercised in one facility. Furthermore, the environment can be shared with other users who are utilizing the same database. Such a facility has great potential for space station assembly or extra-vehicular training. In addition, this would be an ideal facility for virtual training in Lunar or Martian environments.

The Mockup & Trainer Section and the PLAID Lab at JSC began collaborating on a concept study in July 1994, which has become focused on exploring hybrid tracking systems and delivering tracking information over an ethernet network to provide VE capabilities for the Pogo.

CURRENT VE SYSTEM ARCHITECTURE

The hardware components that make up this particular VE system are all commercially available. The present VE hardware capabilities available on the market are adequate for studying this application and for determining the issues which will need to be resolved in order to materialize the concept. Due to the large working volume of this facility (1 meter in width, 2 meters in height, and 10 meters in length) a hybrid system will be tested and evaluated to gather the subject's position information.

Two SGI Crimson workstations are currently being utilized in this system. The platform will soon be upgraded with an Onyx. Besides the graphical workstations, a PC is being used to transmit the I/O from a DC pulsed magnetic tracker and a right handed glove to the scene generator.

The software found in the PLAID Lab, the PLAID/VE version represents some of the most detailed and realistic VE models at JSC. The PLAID Lab has models of the Space Station, the Orbiter , and MIR, with both internal & external views. In addition, it has models of payloads that will fly on the Orbiter through the middle of 1996. The PLAID Lab also has 3-D models of the human body which can be calibrated to an individual's anthropometric characteristics (i.e. height, length of limbs, size, etc.). This model is also known as Jack™ to those who are familiar with this human factors analysis tool. Although Jack™ was conceptually born at JSC, the Center for Human Modeling and Simulation has devoted a great deal of work by developing Jack™ into a fully jointed human figure with real-time movement in three dimensions. EVA tools and foot restraints, which are used by Jack™, are other items at the disposal of the user in the virtual environment.

The first step in the development of the Pogo VR software was to create a data pipeline to connect the Pogo workstation (an IBM PC running Microsoft DOS) to the PLAID Lab's SGI Crimson's. This pipeline would then allow the Pogo computer to send test-subject sensor data to the PLAID Lab computer, which could then generate a graphical image showing the training participant's position and orientation.

Since both the Pogo and PLAID computers were already connected via an ethernet network, the decision was made to implement the data pipeline using sockets.

Thus, the computers would cooperate to create a socket connection, which could then be read and written to byte by byte for sending and receiving data across the network. Sockets are native to UNIX, which made the development of the software for the PLAID Lab computer straightforward. The Pogo computer on the other hand required the purchase of third party programming libraries (from FTP Software, Inc.) to allow the use of sockets with DOS.

Since sensors had not yet been installed on the Pogo computer, the data pipeline software was tested using sensor data which had been recorded in the PLAID Lab. The Pogo computer's data pipeline software read the data from a file and then echoed it across the network. And as the PLAID Lab received the data, an image of a test subject was drawn and updated.

Work is currently underway to modify the Pogo's data pipeline software to read data from real-time sensors, which are now being installed on the Pogo computer.

Trackers

Testing was conducted on the Pogo to determine the tracking capabilities and limitations of a magnetic tracker mounted on the spreader bar of Pogo's gimbal structure. The tracker was on loan at the time and the network data transfer described earlier was not available. Nevertheless, results show that this configuration produces accurate measurements for the head. However, as the distance from the transmitter to the sensors went beyond four feet, poor measurements were being generated. The problems were due to the ferrous materials in the gimbal's bearings which allow the subject to pitch, yaw, and roll. It was determined that additional work was required to reduce or eliminate the ferrous material in Pogo's gimbal to capture useful data with the magnetic tracker so that it could track both the head and hands of a subject in the gimbal.

As tracking solutions were being explored, an idea was developed to utilize two trackers, optical and magnetic, to provide the position and orientation of the head in a large working volume. As it turns out, this idea has already been discussed among researchers such as Biocca [4]. Work is now underway in the PLAID Lab to integrate an optical tracking system with the magnetic tracking system. Essentially, the optical system will be tracking the magnetic transmitter. The relative coordinates of the magnetic tracker will then be determined by its position and orientation with respect to the optical system's point of reference. A demonstration of a passive optical tracking system has been set up to evaluate this scenario. Results will be presented at the 1994 ISMCR Workshop on Virtual Reality.

Although an optical tracking system which provides real-time position AND orientation is not commercially available, vendors are saying that this problem is being worked out and that such a system may be available in the first quarter of 1995. In order to obtain orientation information from current optical trackers, it is necessary to post-process the position data that is taken in real-time.

Concept VE System Architecture

The VE system that may eventually be integrated with the Pogo is shown in Figure 4. PC's are planned to handle all of the I/O for tracking and sound. The number of PC's will have to be determined by the requirements of the various hardware trackers and sensors. The data from the gloves, body suit, and the magnetic tracker will be sent to a serial port in a PC which will then send the information over the ethernet to the scene generator. The latencies will occur at the PC's and at the scene generator. Latencies due to sending the information over the ethernet are not expected. If four serial lines transmit at 19.2k baud, which would equate to approximately 6 kilobytes/second with 25% overhead, this would still be well within the maximum throughput of ethernet, even with additional traffic. Once the information is shipped over the ethernet, latencies will occur as the information is received, reduced (processed), applied to the transformation matrices in the database, and then displayed. The amount of latency is minimal for the current architecture. Only experimentation will be able to determine the amount of latency created by the addition of the other devices. The amount of acceptable latency will also, in turn, determine the maximum number of devices and sensors.

Electromechanical systems, such as the Cyberface 3™, are not being considered because of the limitations to movement and the range involved in the facility. Acoustic systems would be inaccurate in this application due to the amount of acoustical noise inherent in the Pogo and the reflective acoustics of the building which houses the facility. Finally, inertial tracking systems are not being considered because of the inability to recalibrate the individual markers needed, as the errors accumulate, to track the subjects entire body.

Although much work is left to be completed in order to determine the system described above, the VR hardware shown in Figure 4 is planned for integration. A left-handed CyberGlove™ is planned to become a part of the PLAID Lab's VR peripherals. In addition, the Flight Crew Support Division is receiving a Convolvotron™ for the development of spatialized communications during extra-vehicular activity (EVA) in space. Astronauts have great difficulty in determining where another EVA crew member is when they are not within view of the helmet's visor. The 3-D audio communications will assist the astronauts in locating each other as they work in space. The utilization of a body suit to track the entire body of a subject in the Pogo is also being considered. This would allow for the use of Jack™ within the environment where the user could actually look down and see their virtual body within the environment.

CONCLUSION

The experimentation and research that has been conducted thus far has shown that this concept is feasible. The greatest technical hurdle to overcome is the tracking and the latencies of the system. Overhead due to software can be minimized through clever programming to some degree, but hardware latencies will still have to be addressed.

Improvements in the Pogo itself are needed and are currently being addressed, but the application of a virtual environment with this facility will depend on the design and capabilities of the VE System. Overall system latencies will ultimately determine the amount of VR devices and sensors that can be integrated into the Pogo system. This is what is planned to be addressed in research and experimentation as training scenarios are developed for the astronauts.

ACKNOWLEDGMENTS

We extend our appreciation to James Maida, Bennie Matusek, and Abhilash Pandya for their assistance and continuing support.

REFERENCES

[1] Johnson, H. I. and Trader, A. G., Pneumatic Amplifier Controls High Pressure Fluid Supply, NASA Tech Brief, Brief 71-10081, April 1971.

[2] Ray, D. M., Partial Gravity Simulation Using A Pneumatic Actuator With Closed Loop Mechanical Amplification, NASA Technical Memorandum 104798, June 1994.

[3] Burrows, C. R., Fluid Power Mechanisms, 1st ed., London: Van Nostrand Reinhold Co., 1972.

[4] Biocca, Frank A., Hugh Applewhite and Kenneth Meyer, "A Survey of Position Trackers," Presence, MIT, Volume 1, Number 2, 1992, pp 173-200.

[5] Gump, David, Virtual Reality Handbook: Products, Services & Resources, Pasha Publications inc., Arlington, Virginia, 1993.

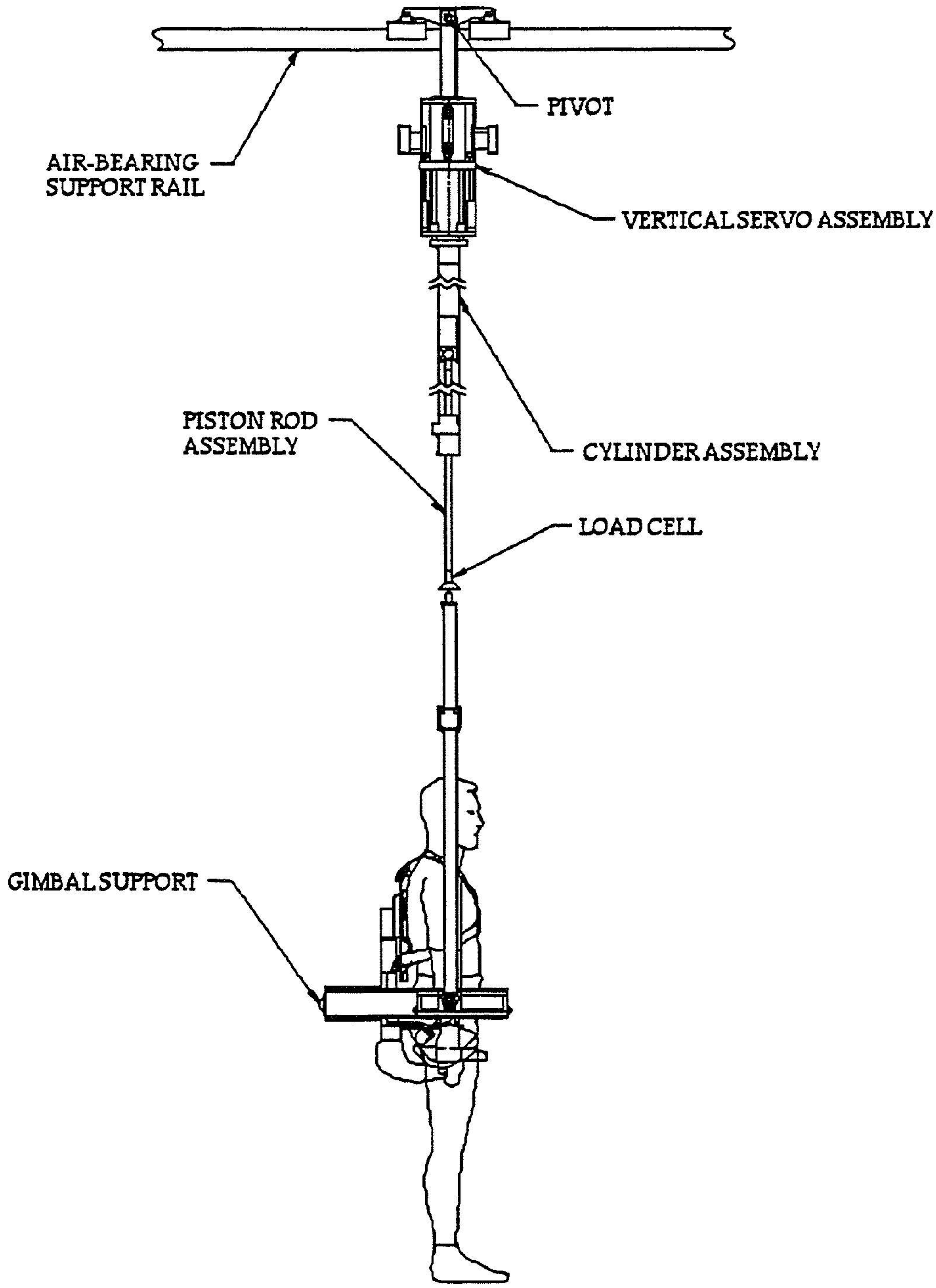


Figure 1. Pogo overall configuration. (Adapted from Trader and Johnson [1] and upgraded by Ray [2]. Gimbal drawn by B. Petty of the Johnson Engineering Corporation.

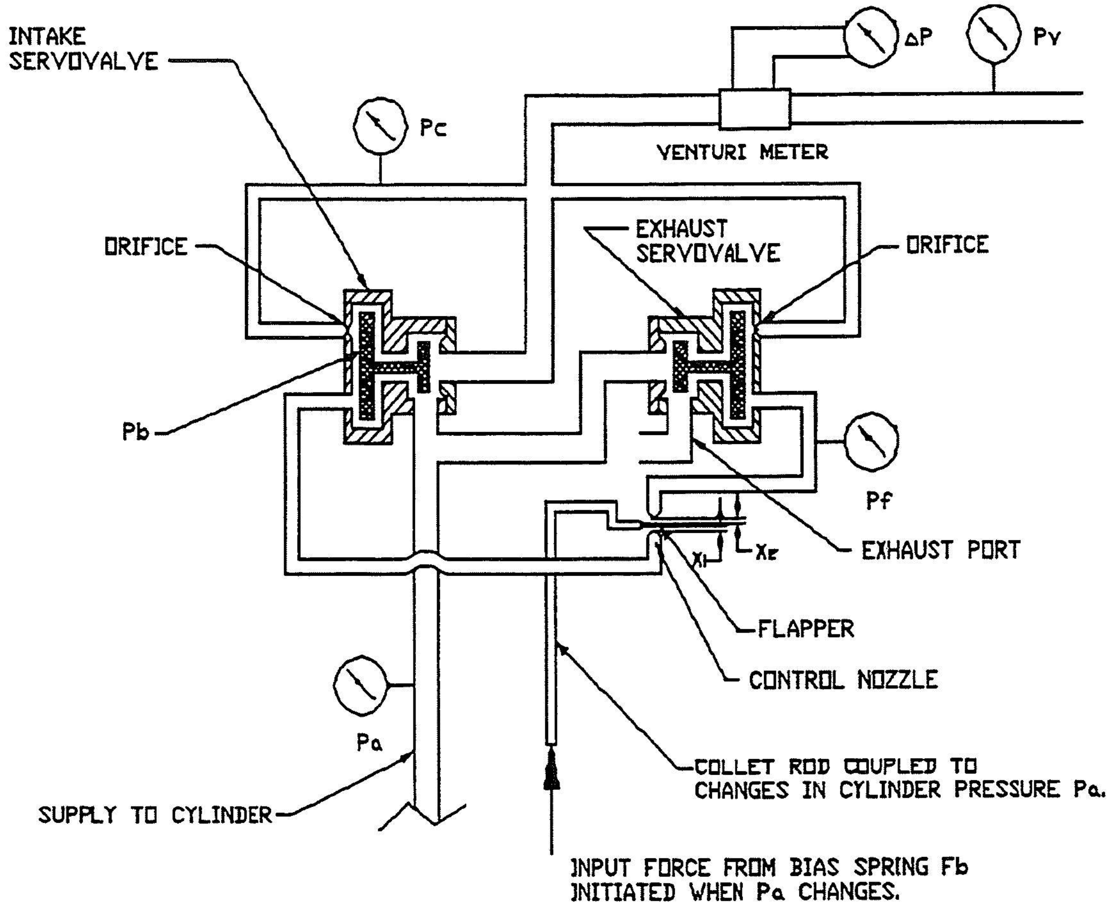
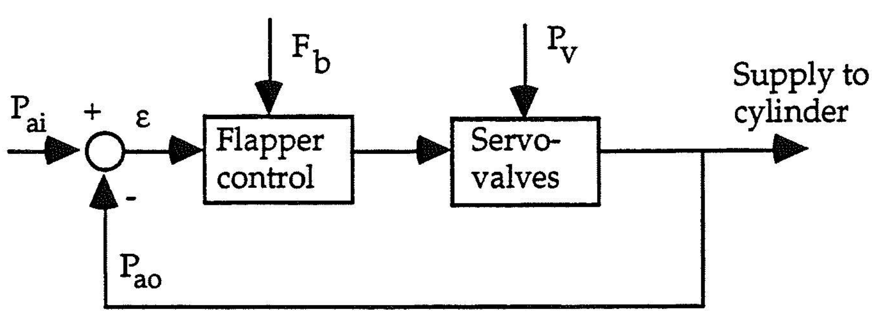


Figure 2. Vertical servo flow control system description.



P_{ai} : Input lifting pressure to the cylinder.

P_{ao} : Instantaneous output to the cylinder.

ε : Error signal.

F_b : Bias spring input force which deflects the flapper.

P_v : Pressure supply to vertical servo.

Figure 3. Control block diagram for the two-stage vertical servo.

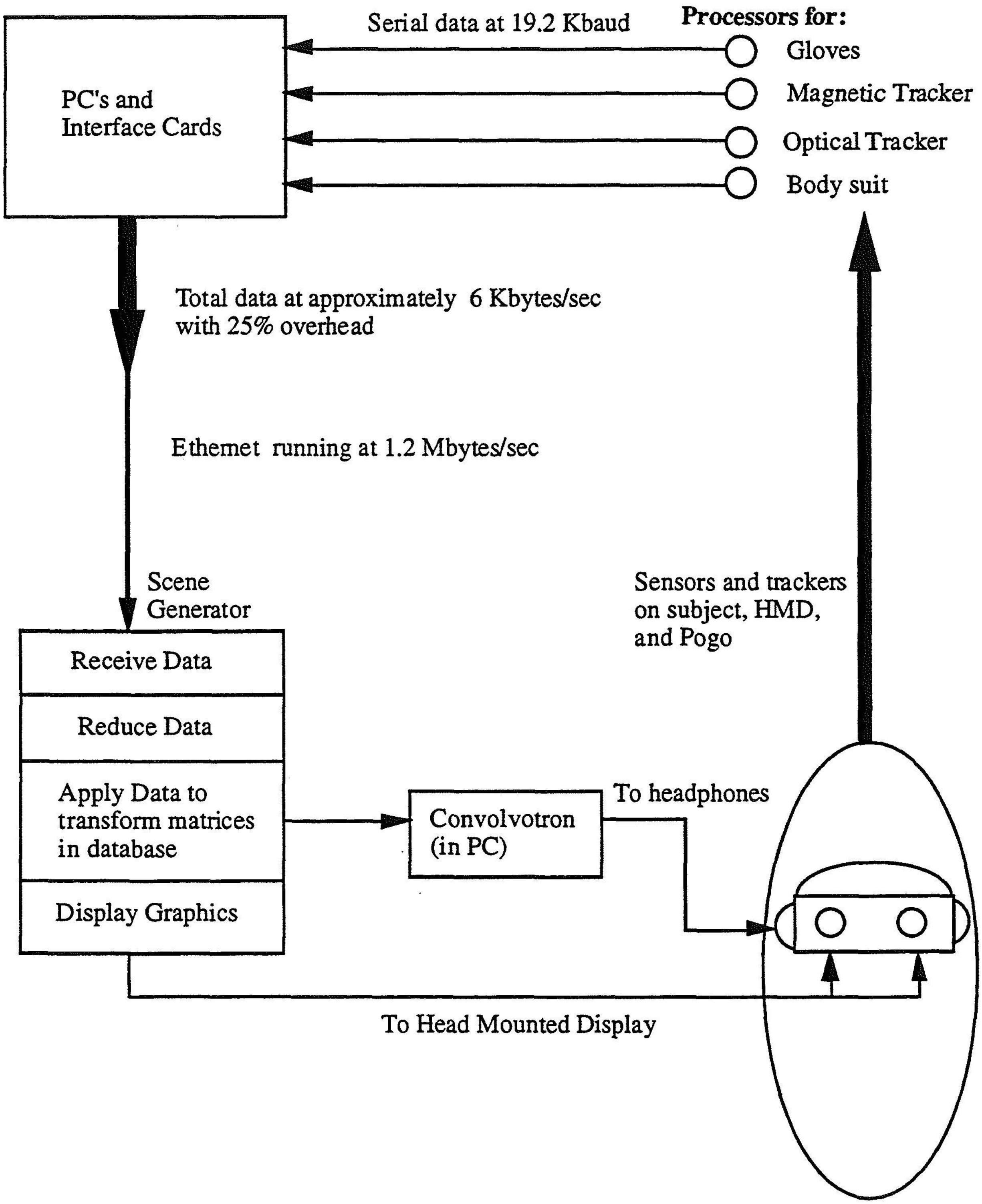


Figure 4. Concept VE System Architechture for Pogo

Session 6
Applications

Applying Virtual Reality to Commercial "Edutainment"

1995109574
N95- 15989
351311
P3

F.Grissom
Sharon P. Goza
S. Michael Goza
Avian Graphics, Ltd.
2111 Castle Drive
League City, TX 77573

Virtual Reality (VR), when defined as a computer generated, immersive, three-dimensional graphics environment which provides varying degrees of interactivity, remains an expensive, highly specialized application, yet to find its way into the school, home, or business. As a novel approach to a theme park-type attraction, though, its use can be justified.

This paper describes how a virtual reality "Tour of the Human Digestive System" was created for the Omniplex Science Museum of Oklahoma City, Oklahoma. The customers main objectives were: 1) To Educate; 2) To Entertain; 3) To Draw Visitors; and 4) To Generate Revenue. The "Edutainment" system ultimately delivered met these goals. As more such systems come into existence the resulting library of licensable programs will greatly reduce development costs to individual institutions.

In order to start the project, Avian Graphics had to first understand what the Omniplex was trying to accomplish with the use of this attraction and VR. They basically wanted to construct a educational fly-through of the human digestive system with the rider having the independent ability to move his head while flying. The educational portion had to meet a wide audience, but mainly concentrated on a younger crowd. The audience was educated using an audio track as well as visual cues. After numerous discussions, several areas were picked as focal points of the tour, the mouth, the stomach, the villi, and the small intestines. Once these were decided upon, a script was developed. The key to the script was to write the audio portion in such a way that children would be able to understand it and adults would still enjoy, be entertained, and possibly even learn something themselves. A professional studio and narrator were employeed to record the script and add appropriate sound effects. This audio recording was then transferred to the graphics system via audio DAT tape to be digitally replayed and synced with the graphics.

To meet the second objective, modifications were made to the audio track and the graphics to provide some excitement. In the mouth you are confronted by flying bits of food while the teeth chomp down around you, while in the stomach, you are tossed about with food particles while sloshing in green acid, and while riding through the lower intestines you are surrounded with roller coaster sounds and travel through dips and turns. Finally with a loud audio sploosh, you are deposited in the toliet and the journey ends.

Omniplex's third objective was met mainly by the use of Virtual Reality. Avian Graphics first had to educate the customer on what VR is truly capable of. The general public has been tainted by Hollywood movies and media fan fare on what VR can do for you. The hard facts are that VR is an emerging technology requiring specialized software and hardware. "Off the shelf" does not exist in the VR world today. Until recently, virtual reality only existed in high dollar research and simulator facilities. Low cost head mounted displays (HMDs), essential for VR, have poor resolutions. The resolution of typical low cost HMDs today is about like taking a Sony Watchman and placing it two inches from your eyes. Also hardware to drive these displays at high frame rates doesn't usually come from your typical PC. They require graphics engines and high speed computers to make it all come to life. After much discussion, Omniplex was convinced that the novelty of virtual reality, even at a lower fidelity because of cost constraints, would draw visitors more than a typical multi–media attraction using real footage but providing less excitement. The popularity of VR with the public provided an instant draw to the attraction. Based on VR arcade attractions, the public was also willing to spend up to a dollar a minute to experience VR. This fulfilled Omniplex's forth objective, to generate revenue to offset the initial investment.

Due to the Omniplex budget constraints, it was finally decided that a medium fidelity simulation of the digestive system would be provided. The hardware was a Silicon Graphics Indigo^2 XL with video card option would provide the compute and graphics power, a Polhemous Fastrack for head tracking, and a helmet mounted display. The original HMD required stereo left and right NTSC video channels which was provided by an IDEN Video Wall. This piece of equipment takes a single NTSC video signal and splits it into four separate quadrants. By placing our images in the proper quadrants of the computer screen, we are able to get the left and right channels required. The video wall also allowed both images to be rendered on one machine thus eliminating the need for expensive multi–headed workstations or synchronous use of multiple machines, thus saving a great deal of expense in both hardware and software. Audio was recorded onto the Indigo^2 and replayed digitally into the headphones of the helmet system also eliminating the need for expensive computer controlled tape recorders. To enhance the images but continue to keep the speed up on the lower end system, all software was custom designed by Avian Graphics and employed several 3D graphics "tricks", but was designed to be 90% reusable for future projects. The total cost of the project was less than $110,000.00 which is a significant price difference from most location based VR entertainment systems currently available to the community.

In conclusion, the Virtual Reality Tour of the Human Digestive System was designed to be a medium resolution, low cost edutainment system. Through discussion of customer objectives, price vs. performance issues, and what was feasible with current technology, the above system met the objectives of the customer and provided an entertaining piece of modern edutainment to capture the interest of today's youth.

1995109575 N95- 15990

A Workout for Virtual Bodybuilders

351312

(design issues for embodiment in multi-actor virtual environments)

P.8

Steve Benford
Department of Computer Science
The University of Nottingham, Nottingham, UK
Tel: 44-602-514203
E-mail: sdb@cs.nott.ac.uk

Chris Greenhalgh
Department of Computer Science
The University of Nottingham, Nottingham, UK
Tel: 44-602-514225
E-mail: cmg@cs.nott.ac.uk

John Bowers
Department of Psychology
The University of Manchester, Manchester, UK
Tel: 44-61-275-2599
E-mail: bowers@hera.psy.man.ac.uk

Dave Snowdon
Department of Computer Science
The University of Nottingham, Nottingham, UK
Tel: 44-602-514225
E-mail: dns@cs.nott.ac.uk

Lennart E. Fahlén
The Swedish Institute of Computer Science
Stockholm, Sweden
Tel: 46-8-752-1539
E-mail: lef@sics.se

ABSTRACT

This paper explores the issue of user embodiment within collaborative virtual environments. By user embodiment we mean the provision of users with appropriate body images so as to represent them to others and also to themselves. By collaborative virtual environments we mean multi-user virtual reality systems which support co-operative work (although we argue that the results of our exploration may also be applied to other kinds of collaborative system). The main part of the paper identifies a list of embodiment design issues including: presence, location, identity, activity, availability, history of activity, viewpoint, actionpoint, gesture, facial expression, voluntary versus involuntary expression, degree of presence, reflecting capabilities, physical properties, active bodies, time and change, manipulating your view of others, representation across multiple media, autonomous and distributed body parts, truthfulness and efficiency. Following this, we show how these issues are reflected in our own DIVE and MASSIVE prototype collaborative virtual environments.

INTRODUCTION

User embodiment concerns the provision of users with *appropriate* body images so as to represent them to others (and also to themselves) in collaborative situations. This paper presents an early theoretical exploration of this issue based on our experience of constructing and analysing a variety of collaborative virtual environments: multi-user virtual reality systems which support co-operative work.

The motivation for embodying users within collaborative systems becomes clear when one considers the role of our bodies in everyday (i.e., non-computer supported) communication. Our bodies provide immediate and continuous information about our presence, activity, attention, availability, mood, status, location, identity, capabilities and many other factors. Our bodies may be explicitly used to communicate as demonstrated by a number of gestural sign languages or may provide an important accompaniment to other forms of communication, helping co-ordinate and manage interaction (e.g., so called 'body language').

In our experience, user embodiment becomes an obviously important issue when designing collaborative virtual environments, probably due to their highly graphic nature, the sense of user immersion, and the way in which designers are given a free hand in creating objects. However, we believe that many of the issues we raise are equally relevant to co-operative systems in general, where embodiment often seems to be a neglected issue (it appears that many collaborative systems still view users as people on the outside looking in). To go a stage further, we argue that without sufficient embodiment, users only become known to one another through their (disembodied) actions; one might draw an analogy between such users and poltergeists, only visible through paranormal activity.

The issue of user embodiment also dominates research into the use of VR in real world simulations which explore how human beings relate to their physical environment. Example applications include ergonomic testing, safety analysis and even the fashion industry (e.g., the recently proposed notion of the Virtual Catwalk [9]). Such applications are primarily concerned with 'realism' in user embodiments, specifically realism in image, proportion or movement, and considerable effort has been invested into detailed modelling of the human body (an excellent discussion of this area and of experiences with the *Jack* system can be found in [10]). Although we might learn a great deal about constructing virtual bodies from this work, we suspect that the goal of realism will be application dependent. For some applications (e.g., simulations), it will be essential; for others (e.g., collaborative information visualization), it seems less pertinent. Indeed, as we shall argue below, our primary goal concerns the identification of key factors in the way we use our bodies when communicating and the representation of these in some efficient (i.e. computationally inexpensive) manner; a goal that seems to point away from the realistic and towards the abstract.

The basic premise of our paper is therefore that the inhabitants of collaborative virtual environments (and other kinds of collaborative system) ought to be directly visible to themselves and to others through a process of direct and sufficiently rich embodiment. The key question then becomes how should users be embodied? In other words, are the body images provided appropriate to supporting collaboration? Furthermore, as opposed to merely discussing the appearance of the virtual body, we also need to focus on its functions, behaviours and its relation to the user's physical body (i.e. how is the body manipulated and controlled?). Thus, an embodiment can be likened to a 'marionette' with active autonomous behaviours together with a series of 'strings' which the user is continuously 'pulling' as smoothly as possible.

Our paper therefore aims to identify a set of design issues which should be considered by the designers of virtual bodies, along with a set of techniques to support them. These are listed in the next section and constitute a diverse, and occasionally conflicting, set of requirements. Designing an appropriate body image will most likely be a case of maintaining a sensible balance between them. Furthermore, this balance may be both application and user dependent and will no doubt be constrained by the available computing resources. In the long term it may be possible to refine our initial list of issues into a body designer's 'cookbook'. However, we do not yet have sufficient experience to do this. Instead, in the final section we describe how the issues are currently reflected in two of our own collaborative virtual environments, DIVE and MASSIVE, and in applications we have developed based on Division Ltd.'s dVS™ system. In each of these cases, we give examples of the bodies we have constructed so far.

DESIGN ISSUES AND TECHNIQUES

In this section we identify a list of design issues for user embodiments as well as possible techniques for dealing with them. As indicated above, we approach these issues from the perspective of collaborative virtual environments, although we encourage the reader to consider their application to other kinds of collaborative system. We begin with the fundamental issues of presence, location and identity.

Presence

The primary goal of a body image is to convey a sense of someone's presence in a virtual environment. This should be done in an automatic and continuous way so that other users can tell 'at a glance' who is present. In a visually oriented system (such as most VR systems) this will involve associating each user with one or more graphics objects which are considered to represent them.

Location

In shared spaces, it may be important for an embodiment to show the location of a user. This may involve conveying both position and orientation within a given spatial frame of reference (i.e., co-ordinate system). We argue that conveying orientation may be particularly important in collaborative systems due to the significance of orientation to everyday interaction. For example, simple actions such as turning one's back on someone else are loaded with social significance. Consequently, it will often be necessary to provide body images with recognisable front and back regions.

Identity

Recognising who someone is from their embodiment is clearly a key issue. In fact, body images might convey identity at several distinct levels of recognition. First, it could be easy to recognise at a glance that the body is representing a human being as opposed to some other kind of object. Second, it might be possible to distinguish between different individuals in an interaction, even if you don't know who they are. Third, once you have learned someone's identity, you might be able to recognise them again (this implies some kind of temporal stability). Fourth, you might be able to find out who someone is from their body image. Underpinning these distinctions is the time span over which a body will be used (e.g., one conversation, a few hours or permanently) and the potential number of inhabitants of the environment (from among how many people does an individual have to be recognised?).

Allowing users to personalise body images is also likely to be important if collaborative virtual environments are to gain widespread acceptance. Such personalisation allows people to create recognisable body images and may also help them to identify with their own body image in turn. An example of personalisation might be the ability to don virtual garments or jewellery. Clearly, this ability might have a broader social significance by conveying status or associating individuals with some wider social group (i.e. cultural and work dress codes or fashions).

Activity, viewpoints and actionpoints

Body images might convey a sense of on-going activity. For example, position and orientation in a data space can indicate which data a given user is currently accessing. Such information can be important in co-ordinating activity and in encouraging peripheral awareness of the activities of others. We identify two further aspects of conveying activity: representing users' viewpoints and representing their actionpoints.

A *viewpoint* represents where in space a person is attending and is closely related to the notion of gaze direction (at least in the visual medium). Understanding the viewpoints of others may be critical to supporting interaction (e.g., in controlling turn-taking in conversation or in providing additional context for interpreting talk, especially when spatial-deictical expressions such as 'over there' or 'here' are uttered). Furthermore, humans have the ability to register the rapidly changing viewpoints of others at a fine level of detail (i.e. tracking the movement of other's eyes even at moderate distances). Previous experimental work in the domain of collaborative three dimensional design has already shown the importance of conveying users' viewpoints [7]. In contrast, an *actionpoint* represents where in space a person is manipulating. Actionpoints typically correspond to the location of virtual limbs (e.g., a telepointer representing a mouse or the image of a hand representing a data glove).

We propose that a user may possess multiple actionpoints and viewpoints. Notice that we deliberately separate where people are attending from where they are manipulating. Although these are often closely related, there appears to be no reason for insisting that they are strictly synchronised; in the real world it is quite possible to manipulate a control while attending somewhere else - indeed, this is highly desirable when driving a car!. Representing actionpoints involves providing an appropriate image of a limb driven by whatever device a user is employing. Representing viewpoint involves tracking where a user is attending and moving appropriate parts of their embodiment. Later on we shall see systems that show general body position, head position or even eye position depending on the power of the tracking facilities in use.

Availability and degree of presence

Related to the idea of conveying activity is the idea of showing availability for interaction. The aim here is to convey some sense of how busy and/or interruptable a person is. This might be achieved implicitly by displaying sufficient information about a person's current activity or explicitly through the use of some indicator on their body. This leads us to the further issue of degree of presence. Virtual reality can introduce a strong separation between mind and body. In other words, the presence of a virtual body strongly suggests the presence of the user when this may not, in fact, be the case (e.g., the mind behind the body may have popped out of the office for a few seconds). This is particularly likely to happen with 'desktop' (i.e. screen-based VR) where there is only a minimal connection between the physical user and their virtual body. This mind/body separation could cause a number of problems such as the social embarrassment and wasted effort involved in one person talking to an empty body for any significant amount of time. As a result, it may be important to explicitly show the degree of actual presence in a virtual body. For example, the system might track a user's idle time and employ mechanisms such as increasing translucence or closing eyes to suggest decreasing presence.

As a concrete example of this issue, we cite some of our early experiences with the DIVE system (see below). One of the interesting aspects of DIVE is that a user process that exits unexpectedly often leaves behind a 'corpse' (an empty graphics embodiment). A long DIVE session may produce several such corpses (particularly when developing and testing new applications), which can cause confusion. As a result, two informal conventions have been established among DIVE users. First, on meeting a stationary embodiment, one grabs it and gives it a shake (DIVE allows you to pick other people up). An angry reaction tells you that the embodiment is occupied. Second, bodies that turn out to be corpses are 'buried' (i.e. moved) below the ground plane. It would be useful to have some more graceful mechanisms for dealing with this problem!

Gesture and facial expression

Gesture is an important part of social interaction and ranges from almost sub-conscious accompaniment to speech to complete and well formed sign languages for the deaf. Support for gesture implies that we need to consider what kinds of 'limbs' are present. Facial expression also plays a key role in human interaction as the most powerful external representation of emotion, either conscious or sub-conscious. Facial expression seems strongly related to gesture. However, the granularity of detail involved is much finer and the technical problems inherent in its capture and representation correspondingly more difficult. A crude, but possibly effective approach, might be to texture map video onto an appropriate facial surface of a body image (e.g., the 'Talking Heads' at the Media Lab [2]). Another approach involves capturing expression information from the human face using an array of sensors on the skin, modelling it and reproducing it on the body image (e.g., the work of ATR where they explicitly track the movement of a user's face and combine it with models of facial muscles and skin [6] and also the work of Thalmann [8] and Quéau[12]).

Voluntary and involuntary expression

This discussion of gesture and facial expression relates to a further issue, that of voluntary versus involuntary expression. Real bodies provide us with the ability to consciously express ourselves as a supplement or alternative to other forms of communication. Virtual bodies can support this by providing an appropriate set of limbs and 'strings' with which to manipulate them. The more flexible the limbs; the richer the gestural language. However, we suspect that users may find ways of gesturing with even very simple limbs. On the other hand, involuntary expression (i.e. that over which users have little control) is also important (looks of shock, anger, fear etc.). However, support for this is technically much harder as it requires automatic capture of sufficiently rich data about the user. This is the real problem we are up against with the facial expression issue - how to capture involuntary expressions.

History of activity

Embodiments might support historical awareness of past presence and activity. In other words, conveying who has been present in the past and what they have done. Clearly we are extending the meaning of 'body' beyond its normal use here. An example might be leaving trails or carving out pathways through virtual space in much the same way as they are worn into the physical world.

Manipulating one's view of other people

In heterogeneous systems where users might employ equipment with radically different capabilities (see MASSIVE below), it will be important for the observer to be able to control their view of other people's bodies. For example, as the user of a sophisticated graphics computer, I may have the processing power to generate a highly complex and fully-textured embodiment. However, this is of little benefit to an observer who does not have a machine with hardware texturing support. Indeed, the complexity of my body would be counter-productive as the observer would be forced to expend valuable computing resource on rendering my body when it could better be used to render other objects. As a result, the observer should be able to exert some influence over how other people appear to them, perhaps selecting from among a set of possible bodies the one that most suits their needs and capabilities. In short, we propose that it is important for the both the owner and the observers of an embodiment to have some control over how it appears.

This requirement poses a serious problem for most of today's multi-user VR systems - that of subjective variability. Current systems are highly objective in their world view. In other words, all observers see the same world (albeit from different perspectives). A notable exception in this regard is the VEOS system [13]. The ability for people to adopt subjective world views (e.g., seeing different representations of an embodiment) represents a significant challenge to current VR architectures.

Representation across multiple media

Up to now we have spoken mainly in terms of visual body images. However, body images will be required in all available communication media including audio and text. For example, audio body images might centre around voice tone and quality, be it that of the real-person or be it artificial. Text body images (as used in multi-user dungeons) might involve text names and descriptions or (in a collaborative authoring application) a text-body's 'limbs' might be represented by familiar word processing tools and icons (cursor, scissors etc.).

Autonomous and distributed body parts

We have discussed virtual bodies as if they are localised within some small region of space. We may also need to consider cases where people are in several places at a time, either through multiple direct presence (e.g., logging on more than once) or though some kind of computer agent acting on their behalf (e.g., issuing a database query while browsing an information visualisation).

Efficiency

There will always be a limit to available computing and communications resources. As a result, embodiments should be as efficient as possible, by conveying the above information in simple ways. More specifically, we suspect that approaches which attempt to reproduce the human physical form in as full detail as possible may in fact be wasteful and that more abstract approaches which reflect the above issues in simple ways may be more appropriate (unless it turns out that users cannot relate to abstract bodies). Furthermore, we need to support 'graceful degradation' so that users with less powerful hardware or simpler interfaces can obtain sufficiently useful information without being overloaded. This suggests prioritising the above issues in any given communication scenario. In fact, the real challenge with embodiment will be to prioritise the issues listed in this section according to specific user and application needs and then to find ways of supporting them within a limited computing resource.

Truthfulness

This final issue relates to nearly all of those raised above. It concerns the degree of truth of a body image. In essence, should a body image represent a person as they are in the physical world or should it be created entirely at the whim or fancy or its owner? We should understand the consequences of both alternatives, or indeed of anything in between. Examples include: truth about identity (can people pretend to be other people?); truth about facial expression (imagine a world full of perfect poker players); and truth about capabilities (this body has ears on, can they hear me?). On the one hand, lying can be dangerous. On the other, constraining people to the brutal physical truth may be too limiting or boring. The solution may be to specify a *gradient* of body attributes that are increasingly difficult to modify. Those that are easy require relatively little resource. Those that are not require more. For example, changing virtual garments might be easy whereas changing size or face or voice might be difficult. Truthfulness may also be situation dependent (i.e. different degrees may be required for different worlds, applications, contexts etc.). For example, as mentioned in the introduction, simulation type VR applications may require a very high level of truthfulness.

In summary, we have proposed a list of design issues that need to be considered by the designers of virtual bodies along with some possible techniques for addressing them. The following section now describes how some of these issues have been dealt with in our own DIVE and MASSIVE prototype collaborative virtual environments.

EMBODIMENT IN DIVE AND MASSIVE

The authors have been involved in the construction of two general collaborative virtual environments, DIVE at the Swedish Institute of Computer Science, and MASSIVE at the University of Nottingham. This section considers how the above design issues are reflected in these systems.

Embodiment in DIVE

Virtual reality research at the Swedish Institute of Computer Science has concentrated on supporting multi-user virtual environments over local- and wide-area computer networks, and the use of VR as a basis for collaborative work. As part of this work, the DIVE

(Distributed Interactive Virtual Environment) system has been developed to enable experimentation and evaluation of research results [5]. The DIVE system is a tool kit for building distributed VR applications in a heterogeneous network environment. In particular, DIVE allows a number of users and applications to share a virtual environment, where they can interact and communicate in real-time. Audio and video functionality makes it possible to build distributed video-conferencing environments enriched by various services and tools.

A variety of embodiments have been implemented within the DIVE system. The simplest are the 'blockies' which are composed from a few basic graphics objects. The general shape of blockies is sufficient to convey presence, location and orientation (the most common example being a letter T shape). In terms of identity, simple static cartoon-like facial features suggest that a blockie represents a human and the ability for people to personalise their own body images supports some differentiation between individuals (DIVE provides a general geometry description language with which users may specify their own body shapes if they wish). A more advanced DIVE body for immersive use texture maps a static photograph onto the face of the body, thus providing greater support for identifying users in larger scale communication scenarios. This body also provides a graphic representation of the user's arm which tracks their hand position in the physical world via a 3-D mouse.

The display of a solid white line extending from a DIVE body to the point of manipulation in space represents actionpoint in a simple yet powerful way and enables other users to see what actions a user is engaged in (e.g., selecting objects). In various DIVE data visualisation applications, each user may also be associated with a different colour which is used to show which data they are accessing (selected objects change to this colour), thereby providing limited peripheral awareness of their activity.

Immersive blockies also support a moving head which tracks the position of the user's head in the real world via their head-mounted display (i.e. a six degrees of freedom sensor attached to the top of the user's head). This is very effective at conveying viewpoint, general activity and degree of presence. Finally, video conferencing participants can be represented in DIVE through a video window.

Figure 1 shows a DIVE conference scenario involving a range of embodiments. From left to right we see: an immersed user with humanoid body, textured face and tracked head and arm; a simple non-immersive blockie sporting a humorous propeller hat; a video conferencing participant; and a second immersive user. The scene also shows some DIVE collaboration support tools: a functioning whiteboard which can also be used to create documents and a conference table for document distribution.

Embodiment in MASSIVE

MASSIVE (Model, Architecture and System for Spatial Interaction in Virtual Environments) is a VR conferencing system which realises the COMIC spatial model of interaction [1][1]. The main goals of MASSIVE are scale (i.e. supporting as many simultaneous users as possible) and heterogeneity (supporting interaction between users whose equipment has different capabilities, who employ radically different styles of user interface and who communicate over an ad hoc mixture of media). MASSIVE has recently successfully been used to demonstrate wide area VR conferencing (between Nottingham and London over the UK's SuperJANET research network).

MASSIVE supports multiple virtual worlds connected via portals. Each world may be inhabited by many concurrent users who can interact over ad hoc combinations of graphics, audio and text interfaces. The graphics interface renders objects visible in a 3-D space and allows users to navigate this space with a full six degrees of freedom. The audio interface allows users to hear objects and supports both real-time conversation and playback of pre-programmed sounds. The text interface provides a MUD-like view of the world via a window (or map) which looks down onto a 2-D plane across which users move. Text users are embodied using a few text characters and may interact by typing messages to one another or by 'emoting' (e.g., smile, grimace, etc.).

The graphics, text and audio interfaces may be arbitrarily combined according to the capabilities of a user's terminal equipment. Furthermore, users may export an embodiment into a medium that they cannot receive themselves (thus, a text user can be made visible in the graphics medium and vice versa). The net effect is that users of radically different equipment may interact, albeit in a limited way, within a common virtual world (e.g., text users may appear as slow-speaking, slow moving flatlanders to graphics users). For example, at one extreme, the user of a sophisticated graphics workstation may simultaneously run the graphics, audio and text clients (the latter providing a map facility and allowing interaction with non-audio users). At the other, the user of a dumb terminal (e.g., a VT-100) may run the text client alone. It is also possible to combine the text and audio clients without the graphics and so on. One effect of this heterogeneity is to allow us to populate MASSIVE with large numbers of users at relatively low cost.

MASSIVE graphics embodiments are similar to DIVE blockies (as with DIVE, users can specify their own geometry via a simple modelling language). Blockies are also automatically labelled with the name of their owner so as to aid identification. In the text interface, users are embodied by a single character (typically the first letter of their chosen name) which shows position and may help identify users in a limited way. An additional line (single character) points in the direction the user is currently facing. Thus, using only two characters, the MASSIVE text

[1] This model, which is not the subject of this paper, provides users with a flexible way of managing communication across multiple media in densely populated virtual spaces via the concepts of aura, awareness, focus, nimbus, adapters and boundaries.

interface attempts to convey presence, location, orientation and identity.

Given MASSIVE's inherent heterogeneity, its embodiments need to convey users' capabilities to one another. For example, considering the graphics interface, an audio capable user has ears; a desk-top graphics user (monoscopic) has a single eye; an immersed stereo user would have two eyes and a text user ('textie') has the letter 'T' embossed on their head. Thus, on meeting another user, it should be possible to quickly work out how they perceive you and through which media you can communicate with them (e.g., should you use the audio channel or send text messages?).

Figure 2 shows an example of the graphics interface and depicts a conference involving five users (we are one of them). We see two non-immersed, audio capable users facing each other across the conference table (ears and a single eye) and a text-only user facing diagonally towards us. We can also see that another non-audio capable user has their back to us.

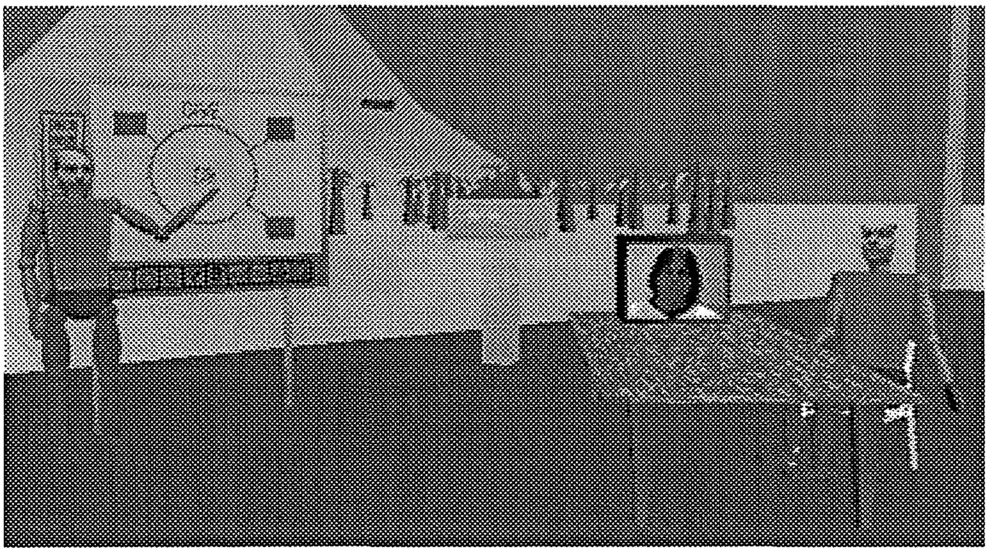


Figure 1: Various embodiments attend a DIVE conference

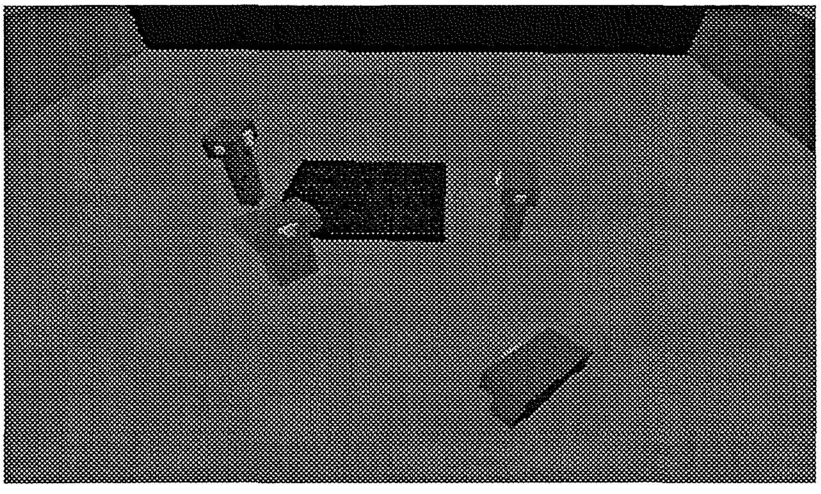


Figure 2: Users show their capabilities at a MASSIVE conference

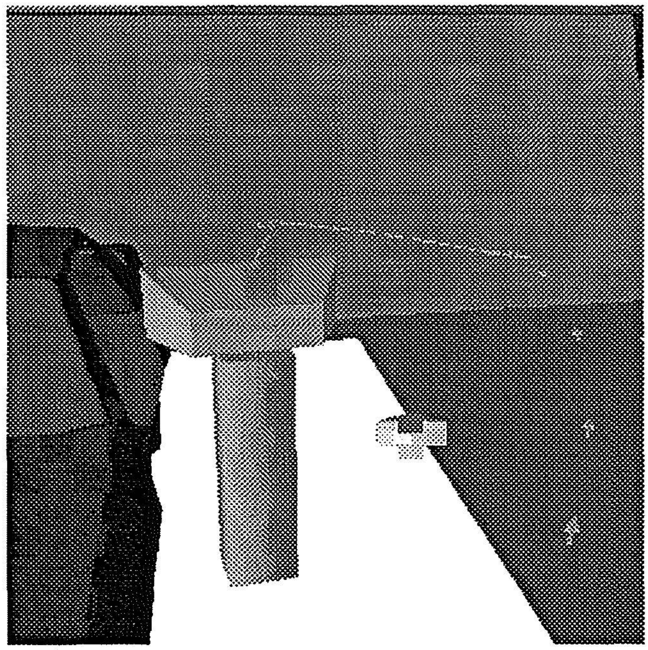


Figure 3: Trails in dVS

Trails in dVS

As part of an on-going UK project into Virtual Organisations called Virtuosi [11], we have begun experimenting with embodiments using the dVS™ VR system from Division Ltd [3]. For background information, Virtuosi is a collaboration between UK academia and industry which will be piloting collaborative virtual environments within two real-world settings, a network of distributed cable making factories where the environments will be used to hold virtual meetings, and the fashion industry, where they will form part of a 'Virtual Catwalk'. Virtuosi partners include the Universities of Nottingham, Manchester and Lancaster, Division Ltd, British Telecom, GPT Ltd, BICC and Nottinghamshire County Council.

Figure 3 shows a screen shot which demonstrates the addition of simple trails to embodiments within dVS in order to convey history of activity. In this case, a person leaves behind a trail of arrows which indicates where they have travelled within their environment. By following these arrows, one can find other people. The trail disappears after a period of time.

SUMMARY

The premise of this paper has been that user embodiment is a key issue for collaborative virtual environments (and indeed, for other kinds of collaborative system). Given this assumption, we have identified the following initial list of issues as being relevant to the embodiment of users: presence, location, identity, activity, availability, history of activity, viewpoint, actionpoint, gesture, facial expression, voluntary versus involuntary expression, degree of presence, capabilities, physical properties, manipulating one's view of others, multiple media, distributed bodies, truthfulness and efficiency. We have also shown how these issues are currently reflected in our own DIVE and MASSIVE prototype collaborative virtual environments.

We suspect that the importance of any given design issue will be both application and user specific and that the art of virtual body building will involve identifying the important issues in each case and supporting them within the available computing resource. However, at the present time, our list remains only an initial framework for the discussion and exploration of embodiment. In our future work we aim to realise a larger number of these issues within our own DIVE and MASSIVE systems, gaining deeper insights into their relative importance and possible implementation. In the longer term, we would hope to refine our list into complete 'body builder's work-out', supporting the choice and analysis of the most appropriate designs for the available equipment, application, users, scale and longevity of intended collaborative applications.

REFERENCES

1. Benford, S., Bowers, J., Fahlén, L. E., and Greenhalgh, C., Managing Mutual Awareness in Collaborative Virtual Environments, *Proc. Virtual Reality Systems and Technology (VRST) '94*, August, 1994, Singapore.
2. Brand, S., The Medialab - Inventing the future at MIT, Viking Penguin, 1987, ISBN 0-670-81442-3, p. 91-93.
3. Grimsdale, C., Supervision - A Parallel Architecture for Virtual Reality, in *Virtual Reality Systems*, Earnshaw, R.A., Gigante, M.A and Jones, H. (eds), Academic Press, 1993, ISBN 0-12-227748-1.
4. Haruo Takemure and Fumio Kishino, *Cooperative Work Environment Using Virtual Workspace*, In Proc. CSCW'92, Toronto, Nov 1992, ACM Press.
5. Lennart E. Fahlén, Charles Grant Brown, Olov Ståhl, Christer Carlsson, *A Space Based Model for User Interaction in Shared Synthetic Environments*, in Proc.InterCHI'93, ACM Press, 1993.
6. Ohya, J., Kitamura, Y., Takemura, H., Kishino, F., Terashima, N., Real-time Reproduction of 3D Human Images in Virtual Space Teleconferencing, *in Proc.VRAIS'93,* IEEE, Seattle Washington September, 1993, pp. 408-414.
7. Shu, L., and Flowers, W., Teledesign: groupware user experiments in three-dimensional computer-aided design, Collaborative Computing, 1(1), Chapman & Hall, 94.
8. Thalmann, D., Using Virtual Reality Techniques in the Animation Process, in *Virtual Reality Systems*, Earnshaw, R.A., Gigante, M.A and Jones, H. (eds), Academic Press, 1993, ISBN 0-12-227748-1.
9. Gray, S.N., The Computerised Catwalk, *Proc. International Conference on Fashion Design*, University of Industrial Arts, Helsinki, May 1993.
10. Badler, N.I., Philips, C.B. and Webber, B.L., Simulating Humans, Oxford University Press, New York, 1993, ISBN 0-19-507359-2.
11. Steve Benford, John Bowers, Stephen Gray, Tom Rodden, Michael Rygol and Vaughan Stanger, The Virtuosi Project, in Proc. VR'94 (London Virtual Reality Expo 1994), Meckler Media, London, February 1994.
12. Quéau, Philippe, Real Time Facial Analysis and Image Rendering for Televirtuality Applications, in Notes from Virtual Reality Oslo '94 — Networks and Applications, eds. Loeffler, Carl E and Søby, Morten and Ødegård, Ola, August 1994.
13. Bricken, William AND Coco, Geoffrey, The VEOS Project, in Presence — Teleoperators and Virtual Environments, Vol. 3, No 2, MIT Press 1994, pp 111--129.

1995 109576

CPP-TRS©: On Using Visual Cognitive Symbols to Enhance Communication Effectiveness

N95-15991

Prof. Graziella Tonfoni, Author of CPP-TRS©
University of Bologna, Istituto di Glottologia
Via Zamboni, 16 - 40126 Bologna, Italy
E-mail: Tonfoni @elet.polimi.it

351315

p.7

Abstract

CPP-TRS© (Communicative Positioning Program/Text Representation Systems) by Graziella Tonfoni is a very easy visual language which is based on a system of 12 canvas, 10 signals and 14 symbols. CPP-TRS© is based on the fact that every communicative action is the result of a set of cognitive processes and the whole system is based on the concept that you can enhance communication by visually perceiving text. Based on a very simple syntax, CPP-TRS is capable of representing meaning and intention as well as communicative function visually. Those are precisely invisible aspects of natural language that are most relevant to getting the global meaning of a text. CPP-TRS is an unambiguous, fast and effective system for reinforcing natural language in human-machine interaction systems. It complements natural language by adding certain important elements that are not represented by natural language in itself. These elements include communicative intention and communicative function of the text expressed by the sender, as well as the role the reader—who is the receiver of the text—is supposed to play. The communicative intention and function of a text and the reader's role are invisible in natural language because neither specific words nor punctuation convey them sufficiently and unambiguously; they are therefore non-transparent. As a language, CPP-TRS can be applied to many different fields both in a transparent and non transparent way.

CPP-TRS© (Tonfoni, 1989-1994) is a visual language that can be productively used to carefully identify and visually represent the sender's communicative intention, the text's function as well as the receiver's role. The CPP-TRS system consists of two consistently integrated parts. CPP stands for Communicative Positioning Program, and it is the methodological component of the system. The CPP methodology enables the user to understand how the sender is positioning himself/herself toward communication. It is a complete program that provides visual schemes, models, and tools aimed toward communicating effectively. TRS stands for Text Representation Systems and is the visual language component. It is strictly integrated with the CPP methodology, and is the corresponding way of representing those cognitive processes and communicative actions, which are being previously identified by CPP. In some ways the sender's communicative intention, the text's function and the receiver's role are more important than words and sentences because they actually control the meaning at a higher level. They are usually apprehended only after processing and interpreting the whole text, which implies time and effort on the receiver's side. In many cases, they are unfortunately missing altogether, because even the writer of the text is not aware of their importance and has no simple means and training to convey them. It also turns out that these elements of text, that are so important in natural language, are also the most difficult to represent in human-machine interaction.

CPP-TRS constitutes a visual representation system that is consistent and not any less significant than the system of punctuation. The punctuation system in written language represents and complements aspects of oral language, such as pauses and intonation, that enable the proper interpretation of texts. There was a time when the punctuation system did not exist, it was actually invented and conventionally accepted to correct the deficiency that occurred when spoken words and sentences were simply written down.

The punctuation system was readily accepted because it was conventionally representing what naturally already existed in oral communication. In written communication pauses and intonation are therefore conveyed by conventional signs such as periods, commas, and question and exclamation marks. These conventional signs have evolved in written language because their antecedents are real. What exists in written language is a complete system of punctuation that is generally shared by all languages, with just some slight variations, that do not compromise the consistency of the overall system.

The CPP-TRS visual language is not any different than the punctuation system since it represents visually those elements, like communicative intention, communicative function and receiver's role, that are invisible but so important. CPP-TRS is both a conventional and natural meta-language that makes explicit from the beginning what otherwise is left to arbitrary interpretation. The fact that a user may make this explicit from the beginning and make it visible is not a constraint to natural language, but a liberating factor. Making language more explicit by adding visual conventions does enhance the final understanding without compromising it.

The punctuation system, among many other things, lets the reader distinguish an interrogative sentence from a declarative one. In a similar way, by reflecting on his/her everyday communicative behavior and being able to identify and use an appropriate visual representation by mastering the CPP-TRS signals and symbols system, the user will be able to recognize his/her communicative intentions and make them explicit to other users.

In human-machine interaction, CPP-TRS can be defined as a communicative traffic control system aimed toward facilitating message production and delivery by pre-interpreting the messages. Like in a language, by knowing the syntax an infinite number of sentences can be generated, and, based on the syntax, processed and understood in a fast, unambiguous and easy way. CPP-TRS allows the user to generate any kind of message, proceeding from very simple instructions toward more complex explanations. By conveying intention and communicative functions visually, interactions occurring in two different languages at the same time can be extremely facilitated.

Musical notation uses a set of visual symbols to convey the composer's intentions and wishes to those performing or executing a composition. These symbols communicate what the notes written on the staff alone cannot.

Text in CPP-TRS is conceived as a musical composition: the receiver "plays" a text, just as a musician executes a composition. Two kinds of symbols are presented. The first kind characterizes the style or type of text. There are eleven of these symbols and they have names, such as describe, define, explain, and so on. The name of each of these symbols has a technical meaning that relates to a cognitive process and identifies a specific intention of text. The second kind of symbols facilitates the interaction between sender and receiver. These symbols —called turn-taking symbols— enable senders and receivers to interpret text more explicitly, and they also indicate immediately when, how, and why the sender wants the receiver to interact. These symbols can be used to direct the turn-taking among senders and receivers of a message, much as a composer uses notation in a composition to direct actions on orchestra members. In order to be able to attribute the right meaning to each of the symbols and signals, any user will need to be trained in cognitive-self awareness, which basically means recognizing what he/she does all the time and being able to match it with the symbols and signals, in the same way that intonation may be represented by the punctuation system.

The whole CPP-TRS system is based on the concept that you can enhance communication by visually perceiving text. The starting point is a global planning and organization of the text, and the ending point is the actual language of the text. This approach is grounded soundly in cognitive research that tries to understand the complexities of how our mind apprehends, processes, and communicates knowledge. The CPP-TRS approach is consistent with theories such as Marvin Minsky's Society of Mind, which contends that many specific cognitive processes occur in our minds before we formulate the actual language of a text. The serious difficulties we encounter organizing a text more often than not are the result of cognitive, not just linguistic, problems.

CPP-TRS directs the user toward starting from a global perspective, reflecting on the intention and function of each text. What is first provided is a set of "canvases", which are visual stimuli and global representations of communicative actions. Canvases are visual schemes that describe various communicative processes themselves, and they are navigation tools to guide the user through the complexities of transmitting knowledge verbally.

Once the user has got a global view of what and how he/she is trying to communicate, he/she can then proceed to a more detailed structuring of text. For this CPP-TRS provides visual signs and symbols.

Signs are visual conventions that represent general types of text. Does, for example, the text give an explanation, or does it summarize something? Does it convey a general concept or is it offered as a comment?

Symbols, on the other hand, have to do with communicative intentions. Is the sender defining something for the receiver or simply describing it? Is the sender trying to explain something or just telling how he/she feels about it?

There are also symbols that facilitate a dialog between the sender and the receiver and they are called "turn-taking" symbols. These symbols explicitly note when the sender wants the receiver to contribute his or her knowledge in developing the text or where the sender is confident that the receiver should just process the text.

CPP-TRS is based on visual aids of canvases, signs, and symbols which are immediate, unambiguous, and consistent, and they can be used in combination.

CPP-TRS starts from a global perspective, reflecting visually on the communicative intention and function of a message (type) as well as the role the receiver should play by getting and returning the message, as it is shown in figure (1).

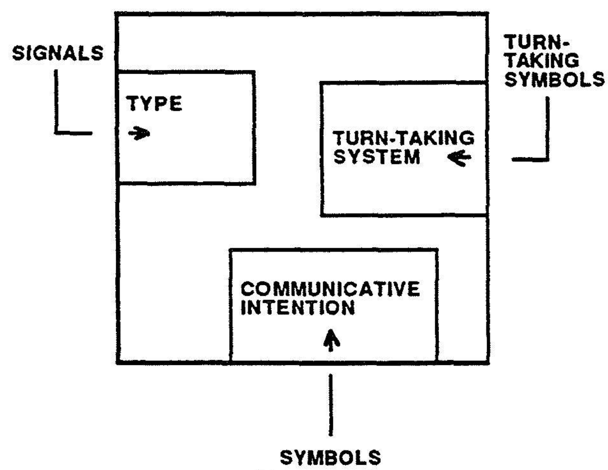


Figure (1)

The CPP-TRS approach differs radically from the traditional approach to organizing text that assumes we create meaning and intention by first stringing together words and sentences.

In CPP-TRS there is no such thing as just language organization and writing, where words and sentence flow out of the mind of the writer and automatically and unambiguously convey his or her exact intention and meaning to the receiver. There is rather a process of communicative labeling that makes transparent the aspects of language that are most difficult to grasp.

Language is actually very complex by itself. For example, it is very difficult to represent that process of understanding which goes from the sentence "It's cold in here. The window is open." to the message "Please close the window." There is no evidence whatsoever for it in the language itself. There is simply no direct link between the words, their meaning, and the sender's intention. This is the kind of ambiguity we find many times in producing and receiving messages, just because it is implicit in language.

In a CPP-TRS perspective the traditional process of creating a message has been reversed. Instead of starting with words, the sender must first understand his/her own intention, next design the structure of the text, and then finally select the words for it. This approach avoids the pitfalls of the traditional approach by making communicative intentions explicit at the very beginning.

The CPP-TRS system thus will support the user with the set of visual tools, shown in figure (2), that are specifically suited to structuring text and communicating effectively.

VISUAL TEXT PERCEPTION

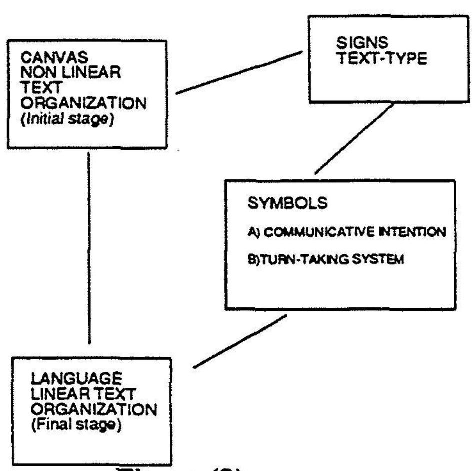


Figure (2)

The same tools can support oral communication very effectively in situations where ambiguity can highly compromise the final result. Air flight control environments are a very well-suited example of such applications.

The system may be used in a way that is either non-transparent or transparent to the reader of the text. The transparent use of the CPP-TRS© system has the visual representation of signs and symbols in the text so that they can be read by the receiver as a visual language complementing natural language. The visual language explicitly conveys those aspects of the text which are implicit and not conveyed linguistically.

More specifically, signs are visual conventions, much as traffic signs. Once they are understood they are easily apprehended, but in themselves they carry little evidence of their meaning.

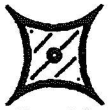
Symbols are also visual conventions, but they visually carry something about their precise meaning so as to reinforce an awareness of the communicative intentions they represent.

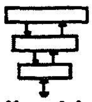
Signs and symbols specifically convey aspects of written communication that cannot be carried by language itself.

Let's now have a look at some CPP-TRS symbols.
There are two kinds of visual symbols: text-styles symbols and turn-taking symbols.

Text-style symbols are specifically aimed toward characterizing the style or type of text. The text-style symbols are:

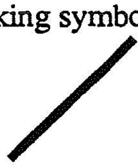
Describe  (from Latin word "describo", which means to write about or write around)
stays for
Organizing information in a free and unconstrained manner. The sender is allowed to provide as much or as little information as he/she chooses without following any logical or chronological order.

Define  (from the Latin word "definio", which means to put limits on)
stays for
Organizing information by restricting it to a selection of relevant information.

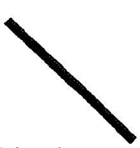
Explain  (from the Latin word "explano", which means to unwrap or open up)
stays for
Organizing information by presenting facts in a cause and effect order. It is possible to start from the original cause and move downward progressively to a set of effects or, alternatively, proceed from the effects and move upward toward the original cause.

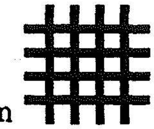
Other text-style symbols are:
narrate
point out
regress
reformulate
syntesize
analyze
express.
They are designed so that they intuitively convey the intention of the text they are associated with.

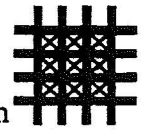
The turn-taking symbols are the following:

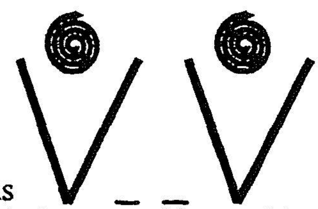
Major scale 

This symbol signals readers that what follows should be read exactly as written.

Minor scale 
This symbol invites readers to modify the marked off portion of the text.

Open or unsaturated rhythm 
This symbol indicates to readers that the writer considers the text to be incomplete. It invites readers to get into that portion of text and add more information if they can.

Tight or saturated rhythm 
This symbol indicates to readers that the writer considers the text as complete.

Vee-like insertions 
The insertion symbols are used in combination with the text-style symbols to explicitly identify the style of text. For example, when used in combination with the describe symbol, they indicate to readers that the portion of text between them is a description.

As it has been illustrated, the CPP-TRS system is aimed toward providing the user with a set of tools for structuring text and communicating effectively. The user can use the system in a way that is either non-trasparent or transparent to the reader of the text. When used non-transparently a writer trained in the system uses visual tools to structure and organize a normal text. The text looks like any normal text to the reader and the reader is not aware and can't see the visual tools the writer used in creating it.

The transparent use of CPP-TRS system on the other side leaves the visual of signs and symbols in the text so that they can be read by the reader as a visual language. The transparent use presupposes that both the sender and receiver have learned the system. A few hours of user training will allow any user to speed up and control any communication process, which may either initiate or respond to.

Conclusions

CPP-TRS is a new paradigm. Numerous contributions made by scholars on iconic language have not being quoted or referred to because the CPP-TRS approach is radically different.
Icons in CPP-TRS are not intended to represent words or sentences as a way to substitute them as in some kind of esperanto as computer-based iconic language research is trying to do; they are rather intended to control different languages at a metalevel.
There are aspects of natural language which icons could never convey or would have problems conveying — one of those many is aspect and time. As the Author of CPP-TRS methodology I took another way: representing visually what natural language does not convey naturally. This is what CPP-TRS icons are designed for.

References

Minsky, Marvin, *The Society of Mind*, Simon & Schuster, 1986.

Schank, Roger, *The Creative Attitude*, Macmillan, 1988.

Tonfoni, Graziella, *La Scrittura multimediale: Tecniche di progettazione e design testuale* (Multimedia Writing: Textual Design and Architecturing), Treviso, Pagus, 1991.

Tonfoni, Graziella, *La Comunicazione aziendale come arte visiva* (Business Communication as Visual Art), Treviso, Pagus, 1991.

Tonfoni, Graziella, *Partitura, solfeggio, movimento: Note di esecuzione di scrittura* (Partition, Solfeggio, Movement: Notes on the Execution of Writing),. Treviso, Pagus, 1992.

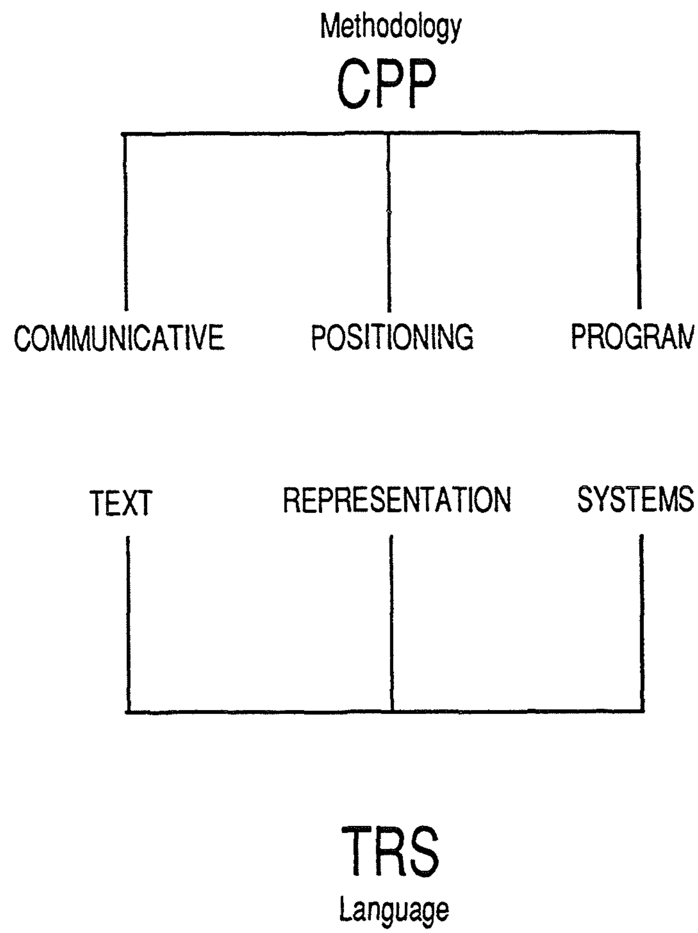
Tonfoni, Graziella (with James Richardson),*Writing as A Visual Art*, Intellect, Oxford, 1994.
(and with a Foreword by Marvin Minsky).

Appendix

CPP-TRS© is both a methodology and a language. Through the methodology (CPP) those invisible aspects of communication are identified and represented by the meta-language (TRS), which complements natural language.

As a language by itself, CPP-TRS© can be applied to a variety of different fields. Here are some applications.

CPP-TRS© by Graziella Tonfoni



CPP-TRS© by Graziella Tonfoni
Applications in Information Processing

- Technical Writing
  - Manual revision and technical writing techniques
  - Client interviewing and information acquisition
  - Information packaging and integration
  - Translation support
- Corporate Communication Training
  - Facilitation and effectiveness of group working sections and meetings
  - Improvement of client presentations
  - Internal training
- Navigation Tools Production
  - E-mail and internal message filtering and delivery
  - Enhancement of software interface design and production
  - Information filing and retrieval

CPP-TRS© by Graziella Tonfoni
Applications in Education

- CPP-TRS© Workshops for Enhancing Writing and Communicative Skills
- CPP-TRS© Based Cognitive Toys
- CPP-TRS© Based Writing Tools

CPP-TRS© by Graziella Tonfoni
Applications in Entertainment

- CPP-TRS© Based Special Effects in Writing Book Previews

  3-D Text Production and Text Motion
- CPP-TRS© Based Writing Paths and Writing Trips
- CPP-TRS© Based Writing Parks and Writing Environments

Scripting Human Animations in a Virtual Environment

Michael E. Goldsby*,
Abhilash K. Pandya*,
James C. Maida

NASA-Johnson Space Center/SP34
*Lockheed Engineering and Sciences Co./C44
2400 NASA Rd. 1
Houston, Tx. 77058
goldsby@graf6.jsc.nasa.gov

Abstract

The current deficiencies of virtual environment (VE) systems are well known; annoying lag time in drawing the current view, environments that are drastically simplified in an effort to reduce that lag time, low resolution and narrow field of view. The scripting of animations is an application of VE technology which can be carried out successfully despite these deficiencies. None of the deficiencies is present in the final product, a smoothly-moving high-resolution animation displaying detailed models. In this animation system, the user is represented in the VE by a human computer model with the same bodily proportions. Using magnetic tracking, the motions of the model's upper torso, head and arms are controlled by the user's movements (18 DOF). The model's lower torso and global position and orientation are controlled by a spaceball and keypad (12 DOF). Using this system the human motion scripts can be extracted from the movements of a user while immersed in a simplified virtual environment. The recorded data is used to define key frames; motion is interpolated between them and post processing is done to add a more detailed environment. The result is a considerable savings in time and a much more natural-looking movement of a human figure in a smooth and seamless animation.

1.0 Introduction

When composing animations portraying moving humans, a way of ensuring natural-looking movements is to capture motion from actual humans [1,2,3,4,5]. Furthermore, placing the person whose movements are being captured in a mockup of the environment which is to be displayed allows registration of position and motion accurately with respect to that environment. We propose the use of a "soft" mockup or a virtual environment (VE) for this purpose.

Human motion can be scripted by specifying individual joint angles or by specifying the goals of the motion and computing the joint angles with an inverse kinematics algorithm [2]. However, the motion produced by both of these methods tends to have an unnatural appearance [6,7,8]. Also, we have found that capturing actual motion takes considerably less time than specifying individual joint angles by interactively specifying movement goals, and produces more realistic motion.

The current deficiencies of VE systems are well known. There are painful tradeoffs between resolution and field of view and between the time it takes to draw the current view and the complexity of the virtual environment [9,10]. Typically one must settle for an unnaturally narrow field of view and a simplified, cartoon-like visual environment. Because the environment in which the motion is captured need only be an approximation of the environment which appears in the final animation, these deficiencies are not a serious hindrance for scripting animations.

2.0 Background

The Graphics Research and Analysis Facility (GRAF) at the Johnson Space Center, Houston, the authors research human modeling as it relates to the human factoring of man-in-the-loop systems. Animations involving human movement are of particular interest for optimizing human performance and for checking consistency and continuity of task designs[11]. Heretofore, the composition of animations involving human movement has been a painstaking operation in which a user at an interactive workstation specifies each movement of each joint. The method of scripting described in this paper results in a considerable savings of time and produces more natural-looking human movements in an animation.

3.0 Description of the system

3.1 Tracking and Computing the Human Motion.

The first phase involves the capture of the tracking information from actual human motion and the computation and display of the resultant motion of the human model within the VE. In order to insure that the models movements are accurate and that its joint angles mimic those of the user, it is necessary for the figure's major anthropometric measurements to be the same as those of the user.

The user wears a head-mounted display (HMD) slaved to the viewpoint by means of a magnetic tracker. The user is personified in the VE as a human model figure with the viewpoint at the figure's eye sites. A total of four trackers suffices to mimic upper-body motion (16 DOF) [1,2,3]; the trackers are positioned on the head, wrists and upper back. The upper-body joint angles are computed with an inverse kinematics (IK) algorithm[6,7,8]. Wrist radial/ulnar deviation is omitted, leaving only 6 DOF for the arm and shoulder making their joint angle computations deterministic; hence the joint angles are rapidly computed and for most motions are constrained to match those of the user. The shoulder complex motion is ignored leading to some error in the motion. Inclusion of the complex clavicle and scapular motion would make the inverse-kinematic computation non-deterministic and difficult to control with one tracker. It is important to note that, in this phase, a simplified VE is sufficient, as long as it contains the visual cues needed for the motion.

The software system is divided into two drawing servers, one reach server, and one magnetic tracking server (See Figure 1). The main client retrieves the current state of the user from the tracking server, polls the spaceball for translation and rotation information, and merges the spaceball information with the tracker information. This information is passed to the reach server which computes the resulting motion in terms of changes in joint angles[12]. The reach server computation is done in a software package called Jack initiated under a NASA university grant by our laboratory at the University of Pennsylvania [6]. The changes in the position and orientation of the figure as well as the joint angle changes of the body are relayed to the drawing servers which update the environment and pipe the needed stereo views to the head mounted display. The advantages of this distributed design is not only speed, but also that any server could reside on any machine on the internet (e.g. tracking information could come from another facility).

The position and orientation of the figure can be controlled by an operator using a six-degree-of-freedom spaceball. Each magnetic tracker matrix is first converted to the coordinate system of the figure (at the base of feet). The spaceball information (relative mode translation and rotation pulses) is accumulated and applied to each of the magnetic tracker matrices in the figure coordinate system. The composite matrices are converted back to global coordinate system to be presented to the inverse kinematic reach server. The scheme allows the figure to be moved by the operator using the spaceball in a natural manner (with respect to the figures coordinate system) while the motions of the user are applied to the human models new translated and rotated coordinate system. The joint angles of the lower limbs can be changed by the operator using the buttons on the spaceball device[1].

3.2 Scripting the Animation.

Scripting the animation involves processing of the captured human motion sequences to produce the key frames of the animation. It requires two people to use the system. The first is the actual personified user with the magnetic trackers appropriately positioned on the body. The second is the operator who will control the position and orientation of the figure in the VE based on the user's requests. The operator will also command the system to write key frames of the animation at appropriate times. The issue of producing an animation that has a realistic time-line is still being researched.

The operator initiates the session by bringing the user to within reaching distance of the specific work environment. The user then performs the activity as prescribed by the task plan. At the operator's signal, the system records the state of every moveable part. The user tells the operator where and how to orient the figure. Upon completion of the session, a file of human motions is produced. These recorded data are used to define key frames; post processing software interpolates motion between the key frames to produce a smooth animation.

3.3 Producing the High Resolution Animation.

The recording of the scripting is done in a simplified VE. Because the post processing is not time-critical, it can use more complex models supplying details that were missing in the VE. The simplified human model is replaced with a high-resolution model and the environment is made much more detailed. The keyfile is then replayed into the animation frame generation program which interpolates between all the key frames. It is also possible to do other special post processing which include texture mapping and realistic lighting (see the section on future work below) (Figure 2).

4.0 Discussion

A narrowed field of view can affect distance judgments adversely [13,14]; however, we found that, within the extent of human reach, it was not difficult to make sufficiently accurate movements. Also, knowing the relative size of objects (i.e. size of hand relative to a workstation screen, for instance) and knowing the approximate location of at least one (your hand) seemed to increase the knowledge of relative distances. One reason may be that stereopsis is a useful distance cue with a person's reach extent [10].

It can be argued that a helmet mounted display is not needed to script the human animations. Scripting an animation using two global views of the human with the user and the operator working the system was tried. When the user tried to view what was being displayed on the monitors, it changed the motion of the human model. There exists an "animation uncertainty principle". That is, the item being measured (the human being) changes as soon as one tries to see one's own changes on a display monitor. In order for a natural looking animation, the user needs to see what they are looking at and working with. It is believed that the more immersed an individual is into the environment, the more realistic the motions will appear. A helmet mounted display provides some of that functionality with some severe limitations.

The user's left and right-eye views can be seen by the spaceball operator on monitors; however, they are not particularly convenient to use when repositioning or reorienting the VE. Hence, a third view is needed which would give the spaceball operator an overview of the action; ideally, the operator should be able to move this viewpoint.

The dramatic effect of realistic motion was caused by very subtle motions. When the user turned her head, there would be slight motions of the waist, and hands. These motions would be very difficult to reproduce manually. When the user looked up, the back would arch by a few degrees and the elbows might swing back.

The spaceball offered a very distinct advantage. The user could stay relatively close to the magnetic tracker source (this is needed for accuracy) and still be "virtually" moved to any location with any orientation within the virtual environment. Moreover, because the HMD and the magnetic trackers have many cables, the user was also safer to stay seated on a chair just moving the head, torso and arms.

With more trackers, we could capture lower body motion also. Walking while tethered with an HMD and magnetic trackers presents some obvious problems. (Perhaps it is fortunate that one does not walk in microgravity.)

5.0 Conclusion

A virtual environment can provide a rapid and convenient way of capturing human motion sequences. Immersion in the virtual environment allows the user to be positioned correctly relative to the environment and to perform accurate reaching movements. A simplified VE can be used to give an adequate display rate for capturing the motion and then replaced by a more detailed environment when the captured motion is used to generate an animation. Other post processing can provide additional special effects in the finished product, a smooth and seamless animation.

6.0 Future Work

Several extensions of this work are planned for the future.

We intend to allow the figure and user to have different bodily dimensions; thus, for instance, we will be able to script movements for the 5th and 95th percentile individuals so beloved of human factors engineers.

A right-handed CyberGlove has already been incorporated into the system. The CyberGlove senses the motions of the joints of the hand (18 DOF). It gives 2DOF for the wrist, supplying the missing wrist radial/ulnar deviation and leaving only 5DOF for the arm and shoulder IK algorithm. Once a left -handed glove is acquired, animations involving both hands will be done.

There is no limit to the amount to the post-processing that can be done once the motion is captured. For instance, the Radiance algorithm is used in the GRAF to do realistic light computations [15]; we would like to use it to provide realistic lighting for the animations. Additional texture maps, or more detailed texture maps, can also be used. If needed, a texture map based recursive animation (animation inside an animation) could be created to reflect, for instance, changing views on a monitor of the Space Shuttle cargo bay operation. This animation could be displayed with texture maps on a monitor within the environment.

Collision detection would be a real convenience in the VE to ensure that the reaches are accurate. Collision detection is computationally expensive, but even a restricted form of it would be useful in the detection of the intersection of one point at the end of the user's extended finger with any of a set of "reachable" objects [16].

It is possible to record the animation with a viewpoint different from the user's, or with a different field of view. One possibility is to allow the viewpoint to move and to specify its position interactively as the animation frames are produced.

Two viewpoints from the recorded data could be reconstructed and used to make a stereo presentation of the animation that could be viewed with the HMD. Synchronization of the two images requires some special measures.

Finally, as soon as we acquire more trackers, we intend to put a second user into a VE.

7.0 Acknowledgments

We would like to thank Dr. Ann Aldridge, Lorraine Hancock, Kim Tran, and Aniece Wheaton for their input.

8.0 References

1. Pandya, Abhilash K, M. Goldsby, J. Maida (1994), "Human Modeling in Virtual Environments", AIAA Conference, May 1994

2. Pandya, Abhilash K, Aldridge, Ann M., and Goldsby, Michael E. (1994), "A comparison of actual human arm motion with an inverse kinematic arm motion computation", presented at the Twelfth Annual Houston Conference on Biomedical Engineering Research, February 10-11, 1994, Houston, TX.

3. Badler, Normal I., Hollick, Michael J., and Granieri, John P. (1993), "Real-time control of a virtual human using minimal sensors", *Presence*, vol. 2, no. 1, Winter 1993, pp. 82-86.

4. Steinfeld, Edward (1989), "Toward artificial users", in *Evaluating and Predicting Design Performance,* ed. Yehuda E. Kalay, pp. 329-346, John Wiley and Sons, New York

5. Doxey, D., Pandya, A., Aldridge, A., Reschke, M. and Maida, J. (1993), "Utilization of a 3D computer human model for the analysis of the change in postural control mechanisms following space flight", *Bioengineering Conference,* Feb. 1993.

6. Zhao, N. Badler (1989), "Real Time Inverse Kinematics With Joint Limits and Spatial Constraints", *MS-CIS-89-09 University of Pennsylvania Technical Report,* 1989.

7. Lee (1982), " Robot Arm Kinematics Dynamics and Control", *IEEE-Computer,* 1982.

8. Korein, James U. (1985), "A Geometric Investigation of Reach", MIT Press, Cambridge, MA, 1985.

9. Chapin, William (1993), "DesignSpace: an exhibit of developing technologies for design", *Computer Graphics: Visual Proceedings,* Annual Conference Series, 1993, pp. 37-38, The Association for Computing Machinery, Inc., New York

10. Thomas, J. C. (1991), "Human factors issues in virtual reality", *Virtual Worlds, Real Challenges: Papers from SRI's 1991 Conference on Virtual Reality,* ed. T. Middleton, pp. 71-75, Mecklen, London.

11. Nguyen, J., Wheaton, A. and Maida, J. (1993), "The PLAID graphics analysis impact on the space programs", August 1993, *SOAR 1993*

12. Goldsby, M., Pandya, A., Aldridge, A., and Maida. J. (1993), "A virtual reality browser for Space Station Freedom", *1993 Conference on Intelligent Computer-Aided Training and Virtual Environment Technology (ICAT-VET-93),* May 5-7, 1993, Houston, TX..

13. Brooks, Frederick P. Jr., (1993), "Virtual reality -- hype and hope: what's real?", Proceedings: IEEE 1993 Symposium on Research Frontiers in Virtual Reality, Oct. 1993, San Jose, CA, p.3.

14. Oyama, E. Tsunemoto, N., Tachi, S. and Inoue, Y. (1993), "Experimental study on remote manipulation using virtual reality", Presence: Teleoperators and Virtual Environments, vol2., no 2, Spring 1994, pp. 24-31.

15. J. Maida, A. Pandya, A. Aldridge(1992), "A Preliminary Comparison and validation of Computer Lighting Simulation Models for Space Applications Using Empirically Collected Data", RTAR Dec. 1992.

16. Hubbard, Philip M. (1993), "Interactive Collision Detection", Proceedings: IEEE 1993 Symposium on Research Frontiers in Virtual Reality", Oct. 1993, San Hose, CA, pp. 112-124

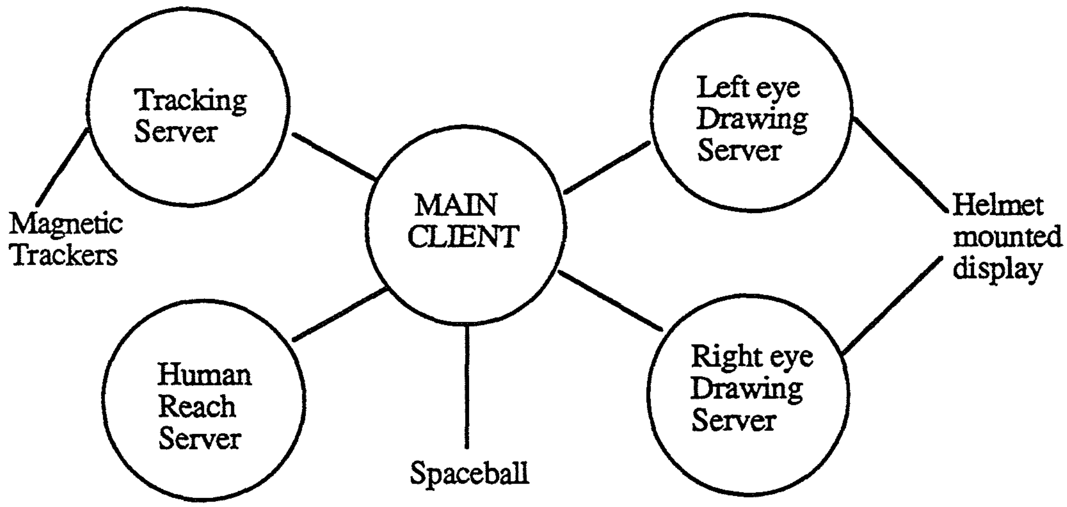


Figure 1- Software/Hardware System Configuration (All servers and clients run on Silicon Graphics Workstations).

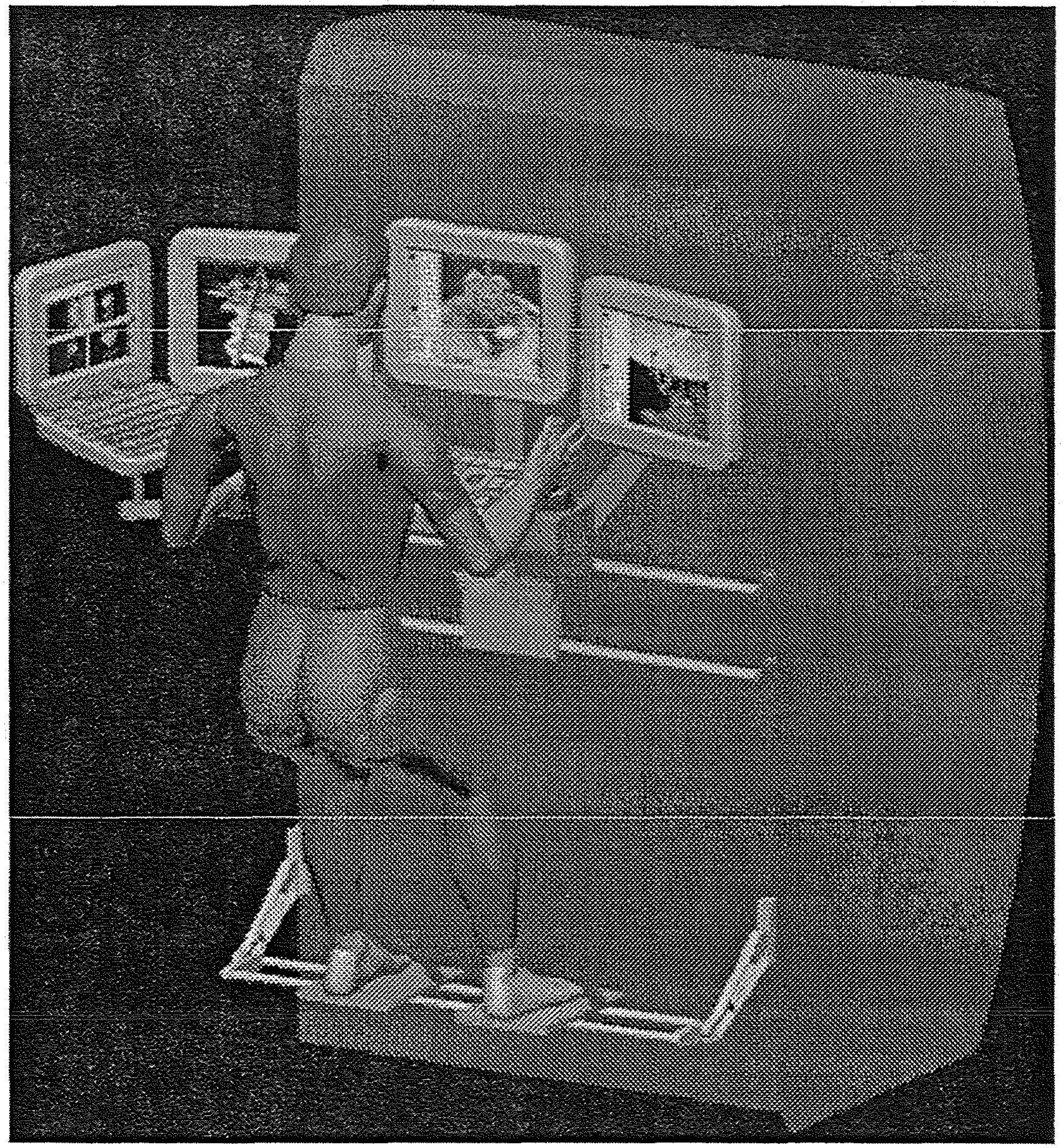


Figure 2. High resolution human model working at a space station workstation.

Author Index

Author Index
(continued)

Author Index
(continued)

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	Nov/94	NASA Conference Publication - Nov. 30 - Dec. 3, 1994

4. TITLE AND SUBTITLE
ISMCR '94: Topical Workshop on Virtual Reality
Proceedings of the Fourth International Symposium on Measurement and Control in Robotics

5. FUNDING NUMBERS

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Lyndon B. Johnson Space Center
Houston, TX 77058

8. PERFORMING ORGANIZATION REPORT NUMBERS
S-788

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
CP-10163

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified/Unlimited
Available from the NASA Center for AeroSpace Information
800 Elkridge Landing Road
Linthicum Heights, MD 21090-2934
(301) 621-0390 Subject category: 63

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Fourth International Symposium on Measurement and Control in Robotics (ISMCR '94) Topical Workshop on Virtual Reality was organized to respond to the growing interest and importance of the niche area of virtual reality in the field of telerobotics and supervised autonomous robotics. The Symposium, organized by IMEKO Technical Committee 17, sponsored by the AIAA/IEEE/ISA, and hosted by the Clear Lake Council of Technical Societies and the NASA Johnson Space Center attempts to bring together a comprehensive international overview of the rapidly moving advanced technology which comprises the field of virtual reality. This focused Symposium deals with each of the critical technology areas in an integrated fashion, such that advances, problems and issues which cut across technologies can be viewed and evaluated from an integrated, common perspective. Papers in the areas of *rendering, tracking sensors, displays, sensory feedback,* and *applications* are included in the six sequential sessions of the Symposium. It is felt that this Symposium provides an important and timely indepth look at the interaction of these technologies as they apply to the applications of virtual reality to robotics.

14. SUBJECT TERMS
virtual reality; artificial intelligence; robotics

15. NUMBER OF PAGES
160

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std. 239-18
298-102