Virtual Reality Calibration and Preview/Predictive Displays for Telerobotics

Won S. Kim
Jet Propulsion Laboratory
California institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

Abstract

A virtual reality (VR) calibration technique of matching a virtual environment of simulated 3-D graphic models with actual camera views of the remote site task environment has been developed. This VR calibration enables high-fidelity preview/predictive displays with calibrated graphics overlay On live video. Reliable and accurate calibration is achieved by operator-interactive camera calibration and object localization procedures based on new linear/nonlinear least-squares algorithms that can handle multiple-camera views. Since the object pose becomes known through the VR calibration, the operator can now effectively usc the semi-automatic computer-generated trajectory mode in addition to the manual teleoperation mode. 'J'redeveloped VR calibration technique and the resultant high fidelity preview/predictive displays were successfully utilized in a recent JPL/NASA-GSFC (Jet Propulsion Laboratory/ Goddard Space Flight Center) telerobotic servicing demonstration. Preview/predictive displays were very useful for both non-contact and contact tasks, providing an effective VR interface with immediate visual prediction/verification to the operator. The positioning alignment accuracy achieved using four camera views in inserting a tool into the ORU hole was 0.51 croon the average with a 1.07 *cm* maximum error at 95% confidence level. Results also indicate that the object localization with two well chosen, e.g., near orthogonal camera views could be nearly as accurate as that with four camera views.

1 Introduction

In telerobotic operation, the local operator site is physically at a distance from the remote site task environment. To cope with this physical separation between the local and remote sites, various techniques on mechanism, sensing, control, machine intelligence, and man-machine interface have been developed over the past several decades (Bejczy, 1980; Stark, et al. 19(S",-; Sheridan, 1992). In particular, a telerobotic system architecture shown in Fig.] employs a VR interface (Ellis, 1994) in the local operator site to cope with this problem. The virtual environment in Fig. 1 tries to simulate the lelnote-silt task environment realistically with a sufficiently natural sensing/control interface to provide the operator with a sense of telepresence or a feeling of being present at the remet.(: site, Examples of VR interfaces towards more realistic sensing/control interfaces include head-mounted television viewing (Goertz, 1965; Liu, Tharp, French, Lai, & Stark, 1993), force reflection (Goertz & Bevilacqua, 1952; Bejczy, Szakaly, & Kim, 1989; Hannaford & Wood, 1989), and anthropomorphic master-slave manipulation (Vykukal, King, & Vallotton, 1973; Tachi, Arai, & Maeda, 1990; Jau, Lewis, & Bejczy, 1994). Although these specific examples aimed at enhancing manual teleoperation do not require modeling/simulation of the task environment, it is in general vital for the VR interface to maintain a simulated virtual model of a remote site task environment in order to support supervisory telerobotic operation, In supervisory telerobotic operation (Sheridan, 1992), the operator issues higher-level commands that can be executed autonomously whenever possible. Autonomous execution of higher level commands generally demands model-based control (Stark, Mills, Nguyen, & Ngo, 1988; Nguyen & Stark, 1989; Tendick, Voichick, Tharp, & Stark, 1991).

Graphic simulation of telerobotic operations based on 3-D geometric models of the remote site task environment has been widely used for off-line task analysis and planning and also for introductory operator training (Kim & Bejczy, 1991). Several commercial graphics packages (for example, Deneb Robotics IGRIP/TELEGRIP, Silma CimStation, Technomatix ROBCAD) are now available for robot simulation. However, use of graphic simulation during the on-line telerobotic operation, for example, as a tool for on-line preview and predictive visualization, has been limited due to the lack of accurate matching between a graphically simulated virtual environment and an actual task environment. This paper describes our recent development of a VR calibration technique that enables a reliable, accurate matching through ol)ermtor-interactive camera calibration anct object localization.

Although there exist many camera calibration (Sutherland, 1974; Ballad & Brown, 1982; Ganapathy, 1984; Tsai, 1987) and object localization (Faugeras & Hebert, 1983; Horn, 1986; Arun, Huang , &

Blostein, 1987; Waker, Shao, & Volz, 1991) methods, our VR calibration specifically addresses calibrated graphics overlay on live video for use in preview/predictive displays. Our VR calibration procedures were only briefly described in previous papers (Kim & Bejczy, 1993; Kim, Schenker, Bejczy, Leake, & Ollendorf, 1993). In this paper, we present mathematical details of the VR calibration in Sections 2 and 3, followed by experimental results of calibration errors in Section 4. We then describe the historical background and our new developments of high-fidelity preview/predictive displays with applications to ground-controlled telerobotic servicing in space under varying communication time delays in Section 5. In Section 6, we briefly describe a successful use of the VR calibration and preview/predictive displays in the ORU (Orbital Replacement Unit) changeout remote servicing demonstration performed in May, 1993, between JPL and NASA GSFC. Finally we describe the conclusion in Section 7.

2 Camera Calibration Using a Robot Arm

Our camera calibration method which is designed for calibrated graphics overlay has three key new features;1) A robot arm itself is used as the calibration fixture, 2) An operator-interactive data entry is adopted to obtain reliable correspondence data. 3) A nonlinear least-squares algorithm combined with a linear least-squares algorithm is employed to obtain accurate camera parameters. In general, camera calibration requires a calibration fixture to determine the camera calibration parameters. Placing a calibration fixture in the actual remote work site is, however, often not so easy. 'I'bus, in our approach, the robot arm itself is used as the calibration fixture, eliminating a cumbersome procedure of using an external calibration fixture.

An operator-interactive methodology is adopted to provide the correspondence information between 3-D graphic model points of the robot arm and 2-1) image points, since it is still difficult for a computer vision system to find correspondence points reliably. Fig. 2 shows the graphical operator interface used during the operator-interactive camera calibration. The solid-shaded 3-D graphics is displayed on the upper left window, and the live (or stored) video picture received from the remote site appears on the lower left window. As the operator clicks 3-]) model points or 2-D image points, their coordinate values appear' on the scrolled list widget of the upper right camera calibration GUI (graphical user interface), When all desired object points and their corresponding image points are entered at different arm poses, the operator can request the system to compute the camera calibration parameters.

There is a standard linear least-squares method that determine the camera parameters (position, orientation, and focal length) for given 6 or more 3-D object points and their corresponding images

in 2-D screen coordinates (Sutherland, 1974; Ballard & Brown, 1982), by assuming that the image formation of the camera can be modeled by a perspective projection of the 3-D world onto the 2-D image plane (ideal pinhole model). There is also a more accurate, elegant two-stage linear least-squares method (Tsai, 1987) that considers nonlinear radial distortion of lens optics, but this two-stage method is not necessary in our graphic overlay applications, since real-time 3-D graphics workstations in general do not support nonlinear projection. Further we assume that 1) the camera optical axis is perpendicular to the image plane, passing through the center of the camera view (Fig. 3), and 2) the viewport of the graphic overlay window in square-pixel resolution coincides with the full size of the camera view. Based on the above assumptions, we first transform the screen coordinates (u_s, v_s) to image plane coordinates (u, v) for each image point:

$$u = (u_s - c_x)/s_y, \tag{1}$$

$$v = (v_s - c_y)/s_y, \tag{2}$$

where the viewport of the graphic overlay window is defined by (min_x, max_x, min_y, max_y) in screen pixel coordinates, for example, (0,646,0,486) for a NTSC-size window', with the image-plane scale and translation factors for the horizontal axis $s_x = c_x = \text{max\_x}/2$ and for the vertical axis $s_y = c_y = \text{max\_y}/2$. We now define a 3 x 4 camera calibration matrix C that relates 3-1) object points and the transformed (re-centered and uniformly scaled) 2-D image points by

$$w \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{C} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \tag{3}$$

where w is an arbitrary scale factor for homogeneous coordinates, and

$$\mathbf{C} = \mathbf{PV} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The camera calibration matrix C defines the field of view angle, position, and orientation of the camera. The effective focal length f which is the distance from the lens center to the image plane (Fig. 3) defines

the perspective projection P, and the vertical field of view angle is related by $fovy = 2 \cdot cot^{-1} f$. The inverse of the 3-D viewing transform V describes the camera position and orientation relative to the robot base frame. Appendix A concisely describes the linear least-squares algorithm to compute the camera parameters from (3) and (4).

The linear least-squares algorithm described in Appendix A produces accurate estimates of camera parameters, when the input correspondence data points are perfect with a well-prepared calibration fixture. However, as the input data become less perfect. for example. due to an imperfect geometric modeling of the robot arm. the linear algorithm does not guarantee the orthonormality of the 3×3 rotation matrix describing the camera orientation. In general, an iterative nonlinear least-squares algorithm must be employed to obtain the least-squares solution that satisfies the orthonormality of the rotation matrix. In the nonlinear algorithm (Appendix B), three angles instead of 9 elements of the rotation matrix are used to represent the rotation. The nonlinear algorithm could, however, diverge or converge to an undesired local minimum, if the initial guess is not sufficiently close to the actual solution. In order to overcome this problem, we devised a new ttvo-step computational procedure: 1) first, obtain an approximate solution by the linear algorithm, and then 2) apply the nonlinear algorithm by using the linear least-squares solution as an initial guess. Once the camera parameters are obtained, the graphic model of the robot arm can be overlaid on the video camera view.

Four views from three cameras were calibrated for the JPL/GSFC remote servicing demonstration: 1) side view, 2) oblique view, 3) overhead wide-angle view, and 4) overhead zoom-in view. In each camera calibration, the operator typically entered about 15 to 30 data points ill total from 3 or 4 different arm poses. Examples of calibrated graphics overlays at four different robot arm poses for the oblique-view camera are shown in Fig. 4.

3 Object Localization

In our VR calibration, the object localization is performed after the camera calibration to determine the pose (position and orientation) of each object. This enables graphic overlay of both the robot arm and the object(s) on live video. Since the object pose becomes known through object localization, the operator can now effectively use the semi-automatic computer-generated trajectory mode in addition to the manual teleoperation mode. Our object localization method has three key new features: 1) An operator-interactive method is adopted to obtain reliable correspondence data. 2) A projection-based linear least-squares algorithm is extended to consider multiple camera views. 3) A nonlinear least-

squares algorithm combined with the extended linear one is employed to obtain an accurate object pose from multiple camera views. The operator's interactive data entry procedure for the object localization is essentially identical to that for camera calibration (see Fig. 2 and Section 2), except that the operator this time enters corresponding points for an object (not for the robot arm) with several different camera views.

Determining the object pose from given 2-D camera views has been studied as one of the key issues in robot vision. Roberts (196.5) showed in his pioneering work that the pose of an object can be conveniently described by a homogeneous coordinate transform, He further showed that the object pose can be determined from a single view of an object by using a least-squares algorithm if 6 or more 3-1) object model points and their corresponding '2-1) image coordinates are available. Sutherland (1974) later developed a more straightforward least-squares algorithm. Both Roberts' and Sutherland's algorithms are projection-based 3-D to 2-D point matching, since 3-D object points are projected onto the 2-J) image plane to match with the actual camera image points.

There are also several object localization algorithms based on 3-D to 3-D point matching (Faugeras & Hebert, 1983; Arun, Huang, & Blostein, 1987; Waker, Shao, & Volz, 1991), when 3-D object points are given in two different coordinates, for example, one in object model coordinates and the other in visual/range sensor coordinates. When stereo camera views are available for an object, both approaches are possible in determining the object pose: 1) an extended projection-based algorithm that does 3-I) to 2-D point matching for both camera views, and 2) a triangulation method that computes the 3-D position from the 2-].) stereo-pair image points by triangulation and perform matching in 3-I) space. Simulation results comparing the two methods indicate that the projection-based method is preferred, in particular, in terms of the orientation estimation accuracy (Kim & Stark, 1989). Another' key advantage of the projection-based approach is that a single image point data corresponding to each object point is acceptable as an input data, while at least two image points corresponding to each object point are required for the triangulation-based approach. Thus. we developed projection-based linear/nonlinear least-squares algorithms extended for multiple camera views.

An image point (u, v) in camera image coordinates and its corresponding object point located at (x, y, z) in object model coordinates are related by

$$w\begin{vmatrix} u \\ v \\ 1 \end{vmatrix} = \mathrm{CM}\begin{vmatrix} x \\ y \\ z \\ 1 \end{vmatrix} \tag{5}$$

where the camera calibration matrix C is given by (4), and the object pose matrix M contains a 3×1 column vector $\mathbf{t}_m$ and a 3×3 rotation matrix $\mathbf{R}_m$, describing the position and orientation of the object model frame, respectively, relative to the robot base frame.

$$\mathrm{M} = \begin{bmatrix} \mathbf{R}_m & \mathbf{t}_m \\ 0 & 1 \end{bmatrix} = \begin{vmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{vmatrix}. \tag{6}$$

When a single camera view of an object and its camera calibration matrix are given for object localization, we can proceed with the linear/nonlinear least-squares algorithms that are basically the same as those of the camera calibration described earlier except that this time the effective focal length f is known. When multiple camera views of an object are given, algorithms for locating the object are computationally more involved as described below.

After some algebraic manipulation of (5)– (6), we can derive a linear least-squares algorithm that determines the object pose for given multiple camera views with known camera parameters. Let $a_i = c_{1i}f + c_{3i}u, b_i = c_{2i}f + c_{3i}v$ for $i = 1,2,3,4$, and $\mathrm{X} = (r_{11}, r_{12}, r_{13}, r_{14}, r_{21}, r_{22}, r_{23}, r_{24}, r_{31}, r_{32}, r_{33}, r_{34})^T$. Then,

$$\begin{bmatrix} a_1x & a_1y & a_1z & a_1 & a_2x & a_2y & a_2z & a_2 & a_3x & a_3y & a_3z & a_3 \\ b_1x & b_1y & b_1z & b_1 & b_2x & b_2y & b_2z & b_2 & b_3x & b_3y & b_3z & b_3 \end{bmatrix} \mathrm{X} = \begin{bmatrix} -a_4 \\ -b_4 \end{bmatrix} \tag{7}$$

Since each pair of object and image points yields two equations of (7), we can have a system of linear actuations $\mathrm{Ax} = \mathbf{b}$, where A becomes a $2N \times 12$ matrix for N pairs of object-image points entered. When $N \geq 6$ from 4 or more non-coplanar object points, the least-squares solution can be obtained from the normal equations $\mathbf{A}^T\mathbf{A}\mathrm{x} = \mathbf{A}^T\mathbf{b}$, where $\mathrm{A}^T\mathrm{A}$ is a 12 x 12 matrix. It turns out, however, that the system matrix A is singular (the rank of matrix A is 11), and the solution in general has the form

$$\mathrm{x} = k\mathbf{x}_h + \mathbf{x}_p, \tag{8}$$

where $\mathbf{x}_h$ is a null-space vector solution of the homogeneous equations $\mathbf{A}\mathbf{x} = \mathbf{0}$, and $\mathbf{x}_p$ is the particular solution of the nonhomogeneous equations $\mathbf{A}\mathbf{x} = \mathbf{b}$. The scale factor k is determined by the unit magnitude constraints of the rotation matrix.

Gaussian elimination with complete pivoting can be applied to the normal equations $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$ to compute $\mathbf{x}_h$ and $\mathbf{x}_p$. Since the rank of matrix $\mathbf{A}$ is 11 or one rank deficient, one zero-pivot occurs during the pivoting procedure. By virtue of the complete pivoting which allows both column and row interchanges, the zero pivot actually occurs at the last column and *last* row in the form of $0 \cdot x = 0$, where x is one of the 12 unknown variables. The variable x is called a free variable (Strang, 1980), since any value of x can satisfy the equation $0 \cdot x = 0$. By setting $x = 0$ and proceeding with the standard back substitution procedure, the particular solution $\mathbf{x}_p$ is obtained. By setting $x = 1$ and proceeding with the back substitution, a null-space vector solution $\mathbf{x}_h$ is obtained. Since the abet'e linear least-squares solution does not guarantee the orthonormality of the rotation matrix, the solution is used as an initial approximate guess for the following nonlinear least-squares algorithm.

in the nonlinear least-squares algorithm, a rotation is represented by three rotational angles instead of 9 elements of the rotation matrix $\mathbf{R}_m$. After some algebraic manipulation of (5)-(6), we can show that each object point (x, y, z) and its corresponding image point (u, v) yield the two equations:

$$F = u + f\frac{X}{Z} = 0, \tag{9}$$

$$G = v + f\frac{Y}{Z} = 0, \tag{lo}$$

where

$$\mathbf{R}_m = Rot(z, \gamma)Rot(y, \beta)Rot(x, \alpha), \tag{11}$$

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \mathbf{R}_m \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{1'2}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} U + r_{14} \\ V + r_{24} \\ W + r_{34} \end{bmatrix} + \begin{bmatrix} c_{14} \\ c_{24} \\ c_{34} \end{bmatrix}. \tag{13}$$

When $N = 3$, there are 6 equations for the 6 unknowns, and Newton's method can be used. When $N \geq 4$, Newton-Gauss method can be used to determine the nonlinear least-squares solution for the object pose. The nonlinear algorithm requires the computation of the Jacobian for each iteration. As an example, the partial derivative of function F with respect to α is derived below by using matrix derivative relations (Shutt, 1960; Lucas, 1963):

$$\partial F/\partial \alpha = K_1(\partial X/\partial \alpha) + K_2(\partial Z/\partial \alpha), \tag{14}$$

where

$$\partial X/\partial \alpha = c_{11}(\partial U/\partial \alpha) + c_{12}(\partial V/\partial \alpha) + c_{13}(\partial W/\partial \alpha), \tag{15}$$

$$\partial Z/\partial \alpha = c_{31}(\partial U/\partial \alpha) + c_{32}(\partial V/\partial \alpha) + c_{33}(\partial W/\partial \alpha), \tag{16}$$

with $K_1 = f/Z$, $K_2 = -f(X/Z^2)$, $\partial U/\partial \alpha = r_{13}y - r_{12}z$, $\partial V/\partial \alpha = r_{23}y - r_{22}z$, $\partial W/\partial \alpha = r_{33}y - r_{32}z$.

The above linear/nonlinear object localization method was applied to locate the ORU in the JPL/GSFC remote servicing demonstration task by using four calibrated camera views. Examples of calibrated graphics overlays for the four camera views after the completion of the VR calibration are shown in Fig. 5. Note that both the robot arm and the object graphics models are now superimposed on the video image. Graphic models can be overlaid in wireframe and/or in solid-shaded polygonal rendering with varying levels of transparency, producing different visual effects to the operator. The hidden lines can also be removed or retained.

4 Experimental Results of Calibration Errors

As shown in Fig. 5, four camera views were used in the JPL/GSFC demonstration to perform an ORU changeout remote servicing task: side view (view 1), oblique view (view 2), overhead wide-angle view (view 3), and overhead zoom-in view (view 4). An operator first calibrated each of the four camera views through the operator-interactive camera calibration procedure, and then determined the ORU pose through the operator-interactive object localization procedure. In order to measure calibration errors of our calibration techniques, an operator performed a complete VR calibration run ten times, each run consisting of four camera calibrations and one ORU object localization. In each camera calibration, the operator commanded the robot arm to move to 3 or 4 pre-defined arm poses and entered 3 to 10 robot arm graphics model points and their corresponding video image points at each robot arm pose. In the object localization, the operator entered correspondence data for the ORU

using the calibrated four camera views. The "true" object location was also obtained by reading the actual joint angles of the robot arm after fully inserting the nut driver tool of the robot arm into the top hole of the ORU. During this insertion, the compliant damping control mode was activated in order not to move the ORU.

The averages and standard deviations of the object localization errors for the ten VR calibration runs are listed in Tables 1 and 2, respectively, showing the effect of using different combinations of camera views on object localization errors. The object localization error was obtained by taking the difference between the estimated object pose obtained by the VR calibration and the "true" object pose. The x, y, and z axes of the ORU object frame were aligned with the vertical, horizontal. and insertion (depth) axes of the ORU hole entrance, respectively. Two kinds of positioning errors in inserting the nut driver tool into the ORU hole were considered: alignment and depth errors relative to the "true" ORU hole position. The maximum (worst) alignment errors at 95% confidence level were computed by $|\mu_{\Delta xy}| + 2\sigma_{\Delta xy}$ from Tables 1 and 2, where $|\mu_{\Delta xy}| = \sqrt{\mu_{\Delta x}^2 + \mu_{\Delta y}^2}$ and $\sigma_{\Delta xy} = \sqrt{\sigma_{\Delta x}^2 + \sigma_{\Delta y}^2}$, and the results are listed in Table 3. Similarly, the maximum (worst) depth errors at. 9.5% confidence level were computed by $|\mu_{\Delta z}| + 2\sigma_{\Delta z}$ from Tables] and 2, and are also listed in 'J'able **3.**

When a single camera view is used for object localization, the object localization error along the camera optical axis tends to be large. When the side view (camera view]) was used only, the alignment error (10. 89 *cm* in Table 3) was larger than the depth error (1.27 cm). By contrast, when an overhead view (camera view 3 or 4) was used, the alignment error (4.20 or 4.57 *cm*) was smaller than the depth error (6.46 or 9.04 *cm*). As a more number of camera views are used for object localization, the object localization error tends to decrease, In Table 3, the maximum positioning alignment error was in the range of 3.2S to 10. S9 *cm* for one camera view, 1.15 to 2.91 *cm* for two camera views, 1.13 to 2.08 *cm* for three camera views, and 1.07 *cm* for four camera views. The maximum depth error was in the range of 1.27 to 9.04 *cm* for one camera view, 0.73 to 5.04 *cm* for two camera views, 0.60 to 2.23 *cm* for three camera views, and 1.37 *cm* for four camera views. Object localization with two well chosen camera views, for example, 1+2, 1+3, or 1+4 (near orthogonal views), was nearly as good as that with four camera views. Object localization with two poorly chosen camera views, for example, 3+4 (similar viewing angles except zoom settings), yielded larger errors.

In our demonstration task, all four camera views were used for the ORU object localization. The positioning alignment error for a tool insertion into the ORU hole using four camera views was 0.51 *cm* on the average (computed by $|\mu_{\Delta xy}| = \sqrt{\mu_{\Delta x}^2 + \mu_{\Delta y}^2}$ from 'J'able 1), with a 1.07 *cm* maximum alignment

error at 95% confidence level (from Table 3). The depth error was 0.65 *cm* on the average, with a 1.37 *cm* maximum depth error at 95% confidence level.

5 Preview/Predictive Displays

Ground control of space robots has potential operational benefits in future space missions. Possible future applications include ground-controlled satellite servicing in space, telescience experiments on space platforms such as the Space Station, and ground-controlled remote assembly/construction work on the Moon or Mars. In such ground-controlled remote operations, however, there is an unavoidable communication time delay. The theoretical lower bound of the round-trip communication time delay imposed by the speed of light is 0.56 s to the low Earth orbit via a geostationary communication satellite, and 2.5 s to the Moon. When the existing NASA communication facilities, are utilized, the round-trip time delay between the ground station and the low Earth orbit is expected to be 4 to 8 s to relay data via several communications satellites and ground stations.

As the communication time delay exceeds beyond 0.5 s. it is increasingly more difficult for the human operator to perform remote manipulation tasks. A well known strategy to cope with time delay is the "move and wait" strategy (Ferrell, 196.5). In this strategy, the operator moves the manipulator a small cl istance and then wait to see what happens before the next move. Two important schemes that enhance telemanipulation task perform ance under communication time delay are predictive display (Bejczy, Kim, & Venema, 1990; Kim & Bejczy, 1991; Sheridan, 1993) and shared compliance control. A recent investigation (Kim, Hannaford, & Bejczy, 1992) based on peg-in-hole teleoperation experiments under O to 4 s time delays indicated that the use of shared compliance control at the remote site is essential for time-d~laycci teleoperation.

h'eves and Sheridan (1984) in the MIT Man-Machine Systems Laboratory pioneered the first predictive display for telemanipulation by using a stick-figure graphics model overlaid on the delayed video picture of the manipulator, In this predictive display, the operator drives the graphic model which responds immediately to the human operator's control command, while the actual video image of the arm responds with time delay, following the graphic model. In effect, the graphic model leads or predicts the actual robot arm motion. The effectiveness of the predictive display technique was demonstrated using simple models of the manipulator arm and simple tasks (Hashimoto, Sheridan, & Noyes, 986; Sheridan, 1993).

We have recently extended this original stiCk-figure-fy])C predictive display to high-fidelity pre-

view/predictive displays. High fidelity is achieved by the VR calibration consisting of operator-interactive camera calibration and object localization procedures, enabling reliable and accurate overlay of graphic models on the live video image of a quasi-static telerobotic task environment. After the VR calibration, the operator can actually perform a remote servicing task with preview/predictive displays, which provides effective visual prediction, verification to the operator for both contact and non-contact tasks.

The operator first generates the simulated robot motion trajectory either by a hand controller or by a computer trajectory generator, and then visually verifies the generated robot motion trajectory through previewing the simulated graphics motion of the robot arm against the actual video camera image of the quasi-static remote-site work scene. Once verified, the recorded motion command is sent to the remote site for actual motion execution. in order to eliminate the problem associated with the varying time delay in data transfer, the robot motion trajectory command is not execute'd until the entire trajectory data are received at the remote site. During the execution, compliance/impedance control can be activated. At the local site, the operator monitors the actual motion execution by observing the returned video image of the robot arm motion. When no contact is involved, the video image of the real arm follows the same trajectory as the simulated preview motion. When contact is involved. however, the final positions of the simulated graphics and actual robot arms can be quite different. For this reason, after the completion of the robot arm trajectory command, the simulated graphics arm is updated with the actual final robot joint angle values. This update eliminates accumulation of small motion execution errors as well as large compensation errors due to the compliance/impedance control. Examples of preview/predictive displays with calibrated graphics overlay during the performance of the JPL/GSFC demonstration task are shown in Fig. 6. Preview/predictive displays are not only useful for space telerobotic applications with time delay but also very useful for terrestrial applications with 110 time delay, such as in disposal and remediation of nuclear waste, for safer and more reliable operations.

To support both manual and supervisory telerobotic operation with preview/predictive displays, a graphical operator interface has been developed by using two Silicon Graphics workstations anti one NTSC video monitor. The primary workstation (IRIS-4D/310 VGX) is used for preview/predictive displays and for various GUIs. The second workstation (IRIS 4D/70GT) is solely used for sensor data display, providing graphical visualization of robot arm joint angles, 6-dof force/torc]uc sensor data, and capaciflector proximity sensor data (Das, ***J992).*** Fig. 7 shows a top-level screen layout on the primary workstation screen during the actual task execution. A calibrated preview/predictive graphics overlay on the live video picture appears on the upper left window, and it also appears on the full screen of a

19-*inch* NTSC monitor for better viewing. A Silicon Graphics VideoLab board allows this simultaneous live video displays. An operator-defined synthetic (virtual) camera view which can be set to any desired viewing angle, position and zoom is displayed on the lower left window.

The graphics/robot control main GUI (lower right window in Fig. 7) allows the operator to issue an individual command interactively. The upper 4 panels are for graphics control such as 1) graphics view selection, 2) view translation, rotation, and zoom, 3) rendering mode selection (wire-frame, solid-shaded, wire-frame with hidden line removal, wire-frame with semi-transparent solid surface model), and 4) video image selection (no video, live video, or stored video image file). The lower 4 panels are for robot control. The first panel allows the operator to define and designate "tag" points (target/(lcstillatioll points indicated by xyz coordinate frames) for the computer-generated trajectory mode. The second panel allows the operator to select a desired cartesian control mode for the hand controller control: world (robot base), tool (end effector), and camera-view referenced control, with a specification of the cartesian control frame origin, for example, at the end effector or at the tool tip. The operator can also set the position and orientation gains of the hand controller motion. The same inverse inverse Jacobian cartesian control algorithms that drive the actual remote robot arm were used to drive the local-site simulated robot arm (Leake, 1991). The third panel allows the operator to record, stop, and play back the hand controller motion trajectory for preview simulation. The serial I/O and trajectory data buffers were effectively used to avoid any missing data in sending the hand controller motion data from the real-time system to the UNIX-based Silicon Graphics workstation at 30 Hz through a 9600-baud serial 1/0 line. The last panel allows the operator to send a remote execution command individually to the remote site.

In performing an actual telerobotic servicing task involving a long sequence of graphics/robot control commands, it is much more efficient to utilize an auto sequence task script than to enter each individual command interactively. The task auto sequence GUI (upper right window in Fig. 7) displays a selected auto sequence script on the scrolled list window, and the current command to be executed is highlighted. The operator can execute the highlighted command by clicking the "step" button. The operator can interrupt the current execution by '(cancel" button, or abort the script completely by "abort>' button. Two types of commands exist: local and remote execution commands. Local execution commands that effect only the local site include Graphics, Video, Camcal, Objloc, Reference_frame, Tag_point, and Object grab/release commands. The remote execution commands currently supported include robot arm motion trajectory (TRAJECTORY), control algorithm selection (INVOKE_ALGORITHM),

sensor data request (REQUEST) such as joint angles and force/torque sensor data, camera selection and set point (CAMERA), robot joint move (GOTO_JOINT), and robot cartesian move commands (GOTO_CART)

6 Telerobotic Servicing Demonstration

In May 1993, the developed VR calibration and high-fidelity preview/predictive displays were successfully utilized in demonstrating a ground-simulated ORU changeout remote servicing task under varying communication time delays of up to several seconds. In this demonstration, JPL acted as the operator site simulating the ground control station, and GSFC, more than 2,500 miles away from JPL, acted as the remote work site with a life-size EP (Explorer Platform) satellite task mock-Lip, a Robotics Research Corporation K- 1607 robot arm, and a Lightweight Servicing Tool (LST; socket driver power tool) mounted at the end of the arm. Two key technologies employed to cope with communication time delay were high-fidelity preview/predictive displays described in this paper and wrist force/torque sensor referenced compliance/impedance control (Leake, ***1991***) implemented at GSFC.

The NASA Select television broadcasting channel was used for a live video link (30 frames/s) from NASA-GSFC to JPL. A TCP/IP socket communication through the Internet computer network was used for a bidirectional command/data link. The round-trip Internet socket communication delay between JPL and NASA-GSFC was measured about 0.1 s on the average, although there were sometimes long time delays (e.g., a IO-minute testing indicated that about 0. S'% of the delays was longer than 0..5 s and about 0.01% was longer than 4 s).

The ORU changeout task scenario used in the remote servicing demonstration had the following sequence. 1) Perform camera calibration. 2) Perform object localization to determine the ORU pose. 3) Move the arm from the start position to a position where the LST tip is about 20 cm in front of the entrance of the hole on the ORU module. 4) Move the LST to the immediate entrance of the hole. .5) insert the 1, S'1'. 6) Latch the LST to the ORU. 7) Turn on the power tool to loosen the screw. 8) Pull out the ORU by 5 cm. 9) Continue to withdraw the ORU so that it is about 15 cm apart from the satellite. 10) Move the ORU to a stow position. 11) Move the ORU back to 15 cm in front of the satellite frame. 12) Align the ORU for insertion. 13) Insert the ORU. 14) Turn on power tool to tighten the screw. 15) Unlatch the LST from the ORU. 16) Pull out the LST to about 20 cm away from the ORU. 17) Finally, move the arm back to the start position.

Steps 5, 6, 7, 8,]3, 14, and 15 were executed autonomously by invoking an appropriate algorithm

using the INVOKE_ALGORITHM command. During these steps the robot arm motion involves actual contact with the task environment, and thus was aided by wrist force/ torque sensor referenced compliance/ impedance control. Steps 3, 4, 9, 10, 11, 12, 16, and 17 were executed by the TRAJECTORY command, where the robot trajectory data was generated either by teleoperation using a hand controller or by a computer automatically for a designated target frame. To facilitate automatic trajectory generation, several fixed target frames (destination /tag points) relative to the ORU were pre-defined and displayed (see Fig. 6). When the ORU pose was determined through the VR calibration (Steps 1 and 2), the poses of these target frames were also determined automatically. In the teleoperation mode, these target frames were merely used as a visual aid to the operator in generating a robot trajectory with a hand controller. In the computer-generated trajectory mode, the operator just designated a target frame, and then the computer generated a straight line trajectory from the current robot pose to the designated frame. The teleoperation mode using a hand controller was helpful for fine alignment to compensate for any errors caused by imperfect modeling and gravity compensation, while the computer-generated trajectory mode was very helpful for global robot motion under a quasi-static telerobotic task environment.

"7 Conclusion

We developed a virtual reality (VR) calibration technique for matching a graphically simulated virtual environment with actual camera views of a remote site task environment. This technique enabled high fidelity preview/ predictive displays with calibrated graphics overlay on live video, providing an effective VR interface for telerobotic servicing. This newly developed technique was successfully utilized in the recent JPL/NASA-Goddard ORU changeout remote servicing demonstration, showing the practical utility of high-fidelity preview/Preclictivc displays combined with compliance control, Currently, an insertion of the VR calibration technique into a commercial graphics software product is on-going through a NASA-Industry Joint Technology Cooperation Task. Future planned work includes semi-automated VR calibration using model-based image processing.

Acknowledgment

This work was carried out at the Jet }'repulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The author' would like to thank A. K. Bejczy, P. S. Schenker, H. Das, E. Paljug, and E. Barlow of JPL, and S. Ollendorf, S.

Leake, E. Cheung, and D. Henry of GSFC for their contributions to make the demonstration successful.

APPENDIX A. Linear Least-Squares Camera Calibration

From (3) and (4), we can show that an object point P located at (x, y, z) in object model coordinates and the corresponding image coordinates (u, v) are related by

$$u = -f\frac{c_{11}x + c_{12}y + c_{13}z + c_{14}}{c_{31}x + c_{32}y + c_{33}z + c_{34}}, \tag{17}$$

$$v = -f\frac{c_{21}x + c_{22}y + c_{23}z + c_{24}}{c_{31}x + c_{32}y + c_{33}z + c_{34}}, \tag{18}$$

Rearranging (17) and (18) into a standard form of the system of linear equations $\mathbf{Ax} = \mathbf{b}$, we have

$$\begin{bmatrix} x & y & z & 1 & 0 & 0 & 0 & 0 & ux & uy & uz & u \\ 0 & 0 & 0 & 0 & x & y & z & 1 & vx & vy & vz & v \end{bmatrix} \mathbf{x} = \mathbf{0}, \tag{19}$$

where the unknown camera parameters are represented by a vector $\mathbf{x} = (c_{11}, c_{12}, c_{13}, c_{14}, c_{21}, c_{22}, c_{23}, c_{24}, c_{31}/f, c_{32}/f, c_{33}/f, c_{34}/f)^T$. Since each object point and the corresponding image point yields the above two equations, N object points and their images generate $2N$ linear equations with the dimension of matrix $\mathbf{A}$ becoming a $2N \times 12$ matrix. Since the equations are homogeneous ($\mathbf{b} = 0$), only 11 equations can be independent (the rank of matrix $\mathbf{A}$ is 11) to have a non-trivial solution other than $\mathbf{x} = \mathbf{0}$. A simple approach to obtaining the non-trivial null-space vector solution is to set one of the variables $c_{34}/f = -1$ and then solve for 11 remaining variables. Namely, (19) can be easily rearranged as $\mathbf{By} = \mathbf{c}$, where y consists of the first 11 elements of x with a constant scale factor k. If $N = 5\frac{1}{2}$, the solution for $\mathbf{y}$ can be determined by the well-known Gaussian elimination method. If $N \geq 6$, the system becomes overdetermined and the least-squares solution can be obtained by

$$\mathbf{y} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{c}. \tag{20}$$

In the actual numerical computation, the normal equations $\mathbf{B}^T\mathbf{By} = \mathbf{B}^T\mathbf{c}$ are directly solved by Gaussian elimination or by the Cholesky method (Lawson & Hansen, 1974) without ever computing the inverse of $\mathbf{B}^T\mathbf{B}$ for computational efficiency. Once the null-space vector solution is obtained, the unity magnitude constraints of the rotation matrix can be used to obtain the camera parameters. The polarity of the scale factor k can be resolved based on the fact that all the object points entered by the operator

are in front of the camera. In the actual implementation, the centroid of all the object points entered (x, y, z) is first computed, and then the polarity is determined so that the centroid is in front of the camera, $z_c = c_{31}x + c_{32}y + c_{33}z + c_{34} < 0$.

Our testing revealed that the above linear algorithm tends to yield poor estimates of c_{31}, c_{32}, and c_{33}, when the object is far away from the camera. This is because the camera view approaches the orthographic projection as the object is farther away from the camera (the denominators of (17) and (18) approach c_{34}), and only 9 parameters among the 12 unknowns can be estimated properly by the linear method. The poor solutions of c_{31}, c_{32}, and c_{33} are thus replaced by using the vector cross product relation $(c_{31}\ c_{32}\ c_{33}) = (c_{11}\ c_{12}\ c_{13}) \times (c_{21}\ c_{22}\ c_{23})$.

APPENDIX B. Nonlinear Least-Squares Camera Calibration

The above linear method uses only 2N linear equations obtained from N object points and their corresponding images without considering 6 additional nonlinear equations imposed by the orthonormality constraint of the rotation matrix (Moffitt & Mikhail, 1980; Horn, 1986). The nonlinear algorithm is necessary to take into account this constraint. It is simpler in the nonlinear algorithm to use 3 rotational angles (roll, pitch, yaw) in stead of the 9 elements of the rotation matrix. From (17) and (18), we can observe that each object point (x, y, z) and its corresponding image point (u, v) yields the two equations:

$$F = u + f\frac{X}{Z} = 0, \tag{21}$$

$$G = v + f\frac{Y}{Z} = 0, \tag{22}$$

where

$$\mathbf{R}_c = Rot(z, \gamma)Rot(y, \beta)Rot(x, \alpha), \tag{23}$$

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \mathbf{R}_c \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{24}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} + \begin{bmatrix} c_{14} \\ c_{24} \\ c_{34} \end{bmatrix} \tag{25}$$

$Rot(x,\alpha)$, $Rot(y,\beta)$, and $Rot(z,\gamma)$ are a rotation α about the x axis, a rotation β about the y axis, and a rotation γ about the z axis, respectively. When $N = 3\frac{1}{2}$, there are 7 equations for the 7 unknowns $\mathbf{x} = (\alpha, \beta, \gamma, c_{14}, c_{24}, c_{34}, f)^T$. When $N \geq 4$, the nonlinear least-squares solution can be obtained by the Newton-Gauss method (Murray, 1972), which is a combination of Newton's method and the least squares method originated by Gauss. The k-th iteration can be described as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\mathbf{J}(\mathbf{x}_k)^T \mathbf{J}(\mathbf{x}_k))^{-1} \mathbf{J}(\mathbf{x}_k)^T \mathbf{F}(\mathbf{x}_k), \tag{26}$$

where $\mathbf{F} = (F_1, G_1, \cdots, F_N, G_N)^T$ for N corresponding-point pairs, and the Jacobian is defined as $\mathbf{J} = d\mathbf{F}/d\mathbf{x}$ ($2N \times 7$ matrix). An efficient Jacobian computation can be derived by using matrix derivative relations (Shutt, 1960; Lucas, 1963). For example, partial derivatives of function F for each corresponding point pair are: $\partial F/\partial\alpha = K_1(c_{13}y - c_{12}z) + K_2(c_{33}y - c_{32}z)$, $\partial F/\partial\beta = K_1 W cos\gamma - K_2(U cos\gamma + V sin\gamma)$, $\partial F/\partial\gamma = -K_1 V$, $\partial F/\partial c_{14} = K_1$, $\partial F/\partial c_{24} = 0$, $\partial F/\partial c_{34} = K_2$, $\partial F/\partial f = X/Z$, where $K_1 = f/Z$, $K_2 = -fX/Z^2$, and $K_3 = -fY/Z^2$. One might try to define new functions by multiplying both (21) and (22) by Z. However, the use of these new functions tends to yield severely under-estimated values (close to 0) for f and c_{34}.

References

Arun, K. S., Huang, "1'. S., & Blostein, S. 1). (1987). least-squares fitting of two 3-I) point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9, 698-700.

Ballard, D. H. & Brown, C. M. (1982). *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall

Bejczy, A. K. (19S0). Sensors, controls, and man-machine interface for advanced teleoperation. *Science*, 208(4150), 1327-1335.

Bejczy, A. K., Kim, W. S., & Venema, S. (1990). The phantom robot: predictive displays for teleoperatiou with time delay. Proceedings *of the IEEE International Conference on Robotics and Automation*, Cincinnati, OH, ***5,16-5.51.***

Bejczy, A. K., Szakaly, Z., & Kim, W. S. (19S9). A laboratory breadboard system for dual-arm teleoperation. *Proceedings of the ?'bird* Annual *Workshop on Space Operations Automation and Robotics (S'0,41/ '89)*, NASA Johnson Space Center, TX, 649-660.

Das, 11. (1992). Capaciflector data display. Jet }'repulsion Laboratory Internal Document Interoffice Memorandum, 3474-92-092.

Ellis, S. R. (1994). What are virtual environments. *IEEE Computer Graphics and Applications*, 17-22.

Faugeras, O. D. & Hebert, M. (1983). A 3-D recognition and positioning algorithm using geometric matching between primitive surfaces. *Proceedings of the 8th International Conference on Artificial Intelligence*, 996-1002.

Ferrell, W. R. (1965). Remote manipulation with transmission delay. *IEEE Trans. on Human Factors in Electronics*, HFE-6, 24-32.

Ganapathy, S. (1984). Decomposition of transformation matrices for robot vision. *Proceedings of the IEEE International Conference on Robotics and Automation*, 130- 139.

Goertz, R. C. & Bevilacqua, F. (19.52), .-\ force-reflecting positional servomechanism. *Nucleonics*, 10(11). 43-45.

Goertz, R. C. (196.5). An experimental head-controlled television to provide viewing for a manipulator operator. *Proceedings of the 13th Remote Systems Technology Conference*, Argonne National Laboratory, 57-60.

Hannaford., B & Wood., L. (1989). Performance evaluation of a 6-Axis high fidelity generalized force reflecting teleoperator. *Proceedings of the NASA Conference on Space Telerobotics*, JPL Pub. 89-7, Pasadena, CA, 87-96.

Hashimoto, T., Sheridan, 'J'. B., & Noyes, M. V. (1986). Effects of predictive information in teleoperation with time delay. *Japanese Joz17-71o1 of Ergonomics*, 22(2).

Horn, B. K. P. (1986). *Robot Vision*. Cambridge, MA: MIT Press.

Jau, B., hi., Lewis, M. A., & Bejczy, A., K. (1994). Anthropomorphic telemanipulation system in terminus cont rol mode. *Proceedings of the Tenth CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators and Ro. Man. Sy. 10*, Warsaw, Poland.

Kim, W. S. & Bejczy, .4. K. (J993). Demonstration of a high-fidelity predictive/preview display technique for telerobotic servicing in space. *IEEE Transactions on Robotics and Automation*. 9(5), 69 S-7'02.

Kim, W. S. & Bejczy, A. K. (1991). Graphics displays for operator aid in telemanipulation. *Proceeding.s of the 1},'1.'1,' International Conference on Systems, Man, and Cybernetics*, Charlottesville, VA, 1059-1067.

Kim, W. S., Hannaford, B., & Bejczy, A. K. (1992). Force-reflection and shared compliant control in operating telemanipulators with time delay. *IEEE Transactions on Robotics and Automation*, 8(2), 176-185.

Kim, W. S., Schenker, P. S., Bejczy, A. K., Leake, S., & Ollendorf, S. (1993). An advanced operator interface design with preview/predictive displays for ground-controlled space telerobotic servicing. *Proceedings SPIE Conference 2057: Telemanipulator Technology and Space Telerobotics*, Boston, MA, 96-107.

Kim, W. S. & Stark, L. (1989). Cooperative control of visual displays for telemanipulation. *Proceedings of the IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, 1327-1332.

Lawson, C. L. & Hanson, R. J. (1974). *Solving Least Squares Problems*. Englewood Cliffs, NJ: Prentice-Hall.

Leake, L. (1991). A cartesian force reflecting teleoperation system. *International Journal of Computers and Electrical Engineering, 17(3),* 133-146.

Liu, A., Tharp, G., French, L., Lai, S., & Stark, L (1993). Some of what. One needs to know about using head-mounted displays to improve teleoperator performance. *IEEE Transactions on Robotics and Automation, 9(5), 63 S-6-18.*

Lucas, J. R. (1963). Differentiation of the orientation matrix by matrix multipliers. *Photogrammetric Engineering*, 29, 70S-715.

Moffitt, F. 11. & Mikhail, E. M. (1980). *Photogrammetry*. New York: Harper and Row.

Murray, W. (1972). *Numerical methods for unconstrained optimization*. London and New York: Academic Press.

Noyes, M.V. & Sheridan, T.B. (19S4). A novel predictor for telemanipulation through a time delay *Proceedings oj the 20th Annual Conference* 071 *Manual Control*, NASA Ames Research Center, Moffett Field, CA.

Nguyen, A. & Stark, L. (1989). 3D model control of image processing. *Proceedings oj the NASA Conference on Space Telerobotics*, JPL Pub. 89-7, Pasadena, CA, 213-222..

Roberts, L.G. (1965). Machine perception of three-dimensional solids. In Tippet, J. T. (Ed.), *Optical and Electro - Optical Information Processing*, brew York: MI '1' Press.

Sheridan, T.B. (1 192). *Telerobotics, Automation, and Human Supervisory Control*, Cambridge, MA: MIT Press.

Sheridan, T.B. (1993). Space teleoperation through time delay: review and prognosis. *IEEE Transactions On Robotics and Automation, 9(5), 592-606.*

Shutt, G. H. (1960). On exact linear equations for the computation of the rotational elements of absolute orientation. *Photogrammetria*, 17(1), 34-37.

Stark, L., Kim, W. S., Tendick, F., Ellis, S, Hannaford, B., et. al. (1987). Telerobotics: Display, Control, and Communication Problems. *IEEE Journal of Robotics and Automation*, RA-3, 67-75.

Stark, L., Mills, B., Nguyen, A. H., & Ngo, H. X. (1988). Instrumentation and robotic image processing using top-down model control. In Jamshidi et al. (eds.), *Robotics and Manufacturing*, New York: ASME Press, 675-682.

Strang, G. (19<S0). *Linear Algebra and Its Applications*. Orlando, FL: Academic Press.

Sutherland, I. E. (1974). Three-dimensional data input by tablet. *Proceedings of IEEE*, 62(4), 453-461.

Tachi, S., Arai, 11., & Maeda, T. (1990). Tele-existence master slave system for remote manipulation. *Proceedings of the IEEE International Workshop on intelligent Robotics and Systems (IROS'90)*, Tsuchiura, Japan, 3-13 -31S.

Tendick, F., Voichick, J., Tharp, G., & Stark, L. (1991). A supervisory telerobotic control system using model-based vision feedback. Proceedings *of the IEEE International Conference on Robotics and Automation*, Sacramento, CA.

Tsai, R. Y. (1987). A versatile camera calibration technique for high-accuracy 3D vision metrology using ofl-the-self TV cameras and lenses, *IEEE Journal of Robotics and Automation*, RA-3(4), *3'23-341.*

Vykukal, H. C., King, R. F., & Vallotton, W. C. (1973). An anthropomorphic master-slave manipulator system. In E. Heer (Eds.), *Remotely Manned Systems: Exploration and Operation in Space*, California Institute of Technology, Pasadena, CA, 199-20.5.

Walker, M. W., Shao, 1,., & Volz, R. A. (1991). Estimating 3-D location parameters using dual number quaternions. *CVGIP: Image Understanding*, ***M(3), 358-367.***

camera view(s)	$\mu_{\Delta x}$ (cm)	$\mu_{\Delta y}$ (cm)	$\mu_{\Delta z}$ (cm)	$\mu_{\Delta\theta_x}$ (°)	$\mu_{\Delta\theta_y}$ (°)	$\mu_{\Delta\theta_z}$ (°)
1	1.43	5.24	-0.23	-0.30	1.33	-0.58
2	0.14	0.02	-1.02	0.59	1.54	0.82
3	-1.66	-1.42	3.78	-0.40	1.46	0.43
4	-1.29	.'].64	4.62	-0.03	0.35	0.30
1+2	1. -0.2.5	-0.19	0.38	-0.26	1.45	0]:)-
1+3	I -0.47	I -0.36	1.25	-0.17	1.30	0.li-
1+4	-0.30	0.67	0.13	-0.81	"0.98	0.33
2+3	-0.80	-0.88	0.78	0.51	1.15	0.57
2+4	-0.03	1.31	-1.26	-0.58	1.12	0.17
3+4	-0.88	-0.65	2.64	-0.38	0.89	0.23
1+2+3	-0.52	-0.52	0.83	-0.01	1.32	i 0.30
1+2 +4	-0.25	0.73	-0.10	-0.54	1.15	-0.07
1+3+4	-0.49	0.17	1.00	-0.44	1.06	0.02
2+3+4	-0.50	0.16	0.89	-0.14	1.09	0.16
1+2+3+4	-0.41	0.30	0.65	-0.28	1.17	0.06

Table 1. Object localization error averages, showing the effect of using different combinations of camera views on object localization

camera view(s)	$\sigma_{\Delta x}$ (cm)	$\sigma_{\Delta y}$ (cm)	$\sigma_{\Delta z}$ (cm)	$\sigma_{\Delta\theta_x}$ (°)	$\sigma_{\Delta\theta_y}$ (°)	$\sigma_{\Delta\theta_z}$ (°)
1	0.78	2.62	0.52	0.31	0.25	0.48
2	1.05	1.56	1.20	0.32	0.28	0.28
3	0.58	0.83	1.34	0.69	0.31	0.20
4	0.46	1.15	2.21	0.41	0.52	1.29
1+2	0.22	0.69	0.21	0.33	0.17	0.30
1+3	0.44	0.60	0.26	0.42	0.22	0.16
1+4	0.10	0.18	0.30	0.26	0.13	0.36
2+3	0.51	0.69	0.78	0.28	0.17	0.13
2+4	0.09	0.18	0.33	0.19	0.18	0.33
3+4	0.26	0.67	1.20	0.32	0.21	0.20
1+2+3	0.38	0.55	0.32	0.32	0.13	0.13
1+2+4	0.09	0.16	0.25	0.23	0.10	0.28
1+3+4	0.14	0.28	0.42	0.31	0.12	0.19
2+3+4	0.16	0.39	0.67	0.24	0.14	0.16
1+2+3+4	0.13	0.25	0.36	0.26	0.08	0.16

Table 2. Object localization error standard deviations, showing the effect of using different combinations of camera views on object localization

camera view(s)	alignment error (cm) $\lvert\mu_{\Delta xy}\rvert + 2\sigma_{\Delta xy}$	depth error (cm) $\lvert\mu_{\Delta z}\rvert + 2\sigma_{\Delta z}$
1	10.89	1.27
2	3.28	3.42
3	4.20	6.46
4	4.57	9.04
1+2	1.75	0.80
1+3	2.07	1.17
1+4	1.15	0.73
2+3	2.91	2.34
2+4	1.71	1.92
3+4	2.53	5.04
1+2+3	2.08	1.47
1+2+4	1.13	0.60
1+3+4	1.14	1.84
2+3+4	1.36	2.23
1+2+3+4	1.07	1.37

Table 3. Object localization maximum positioning errors at 95% confidence level

Figure Captions

Fig. 1. A typical telerobotic system with a virtual environment interface.

Fig. 2. Graphical operator interface during the camera calibration.

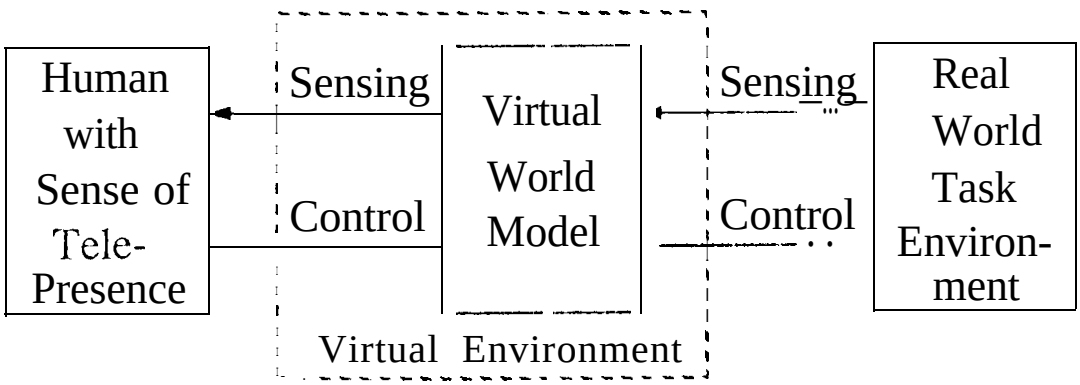
Fig. 3. Camera image formation geometry described by perspective projection. The center of projection is located at the camera lens center. To avoid the reversal of the image, the image plane is placed in front of the lens.

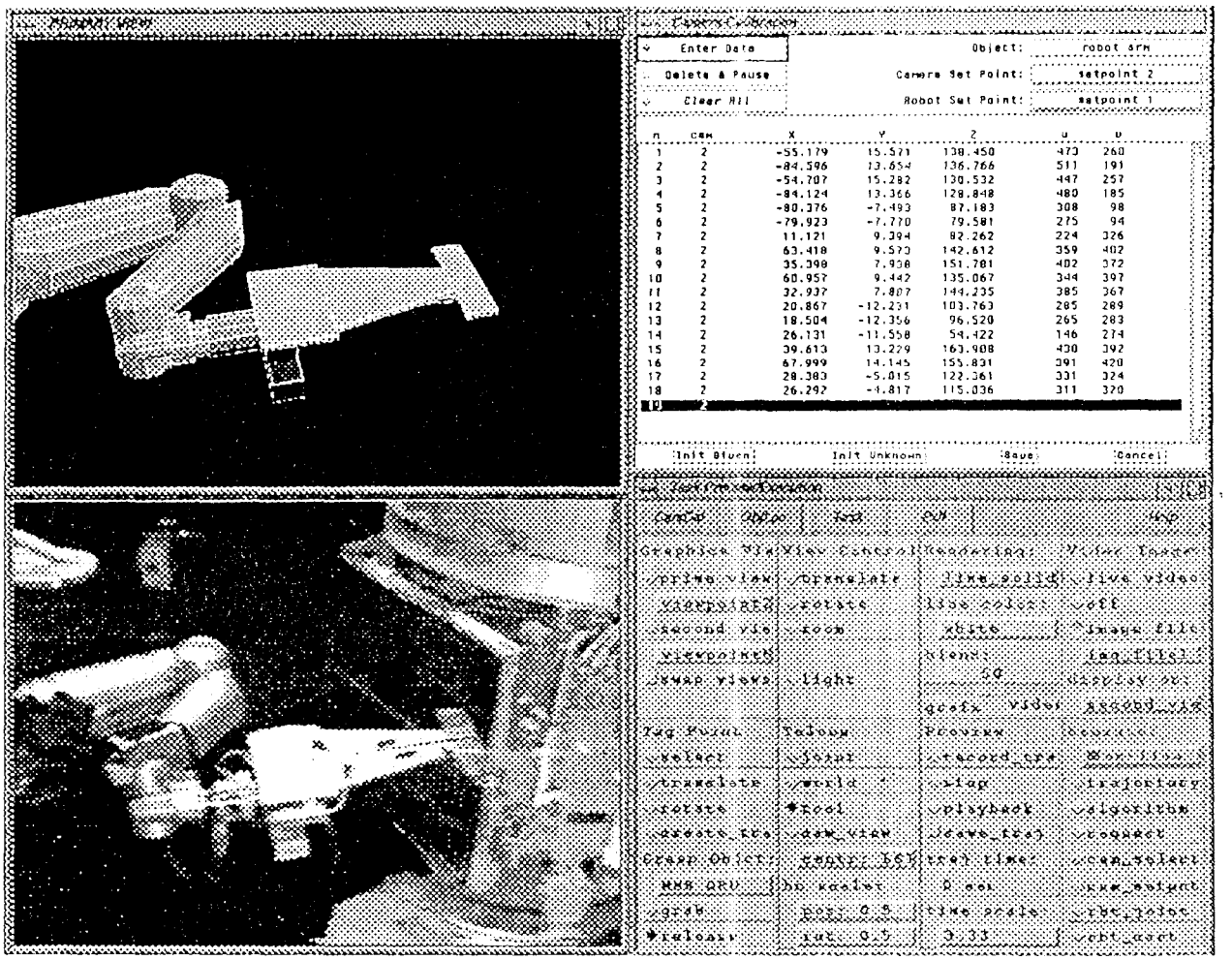
Fig. 4. Calibrated overlays of the robot arm graphics model on the live video picture after the camera calibration at four different robot arm poses.

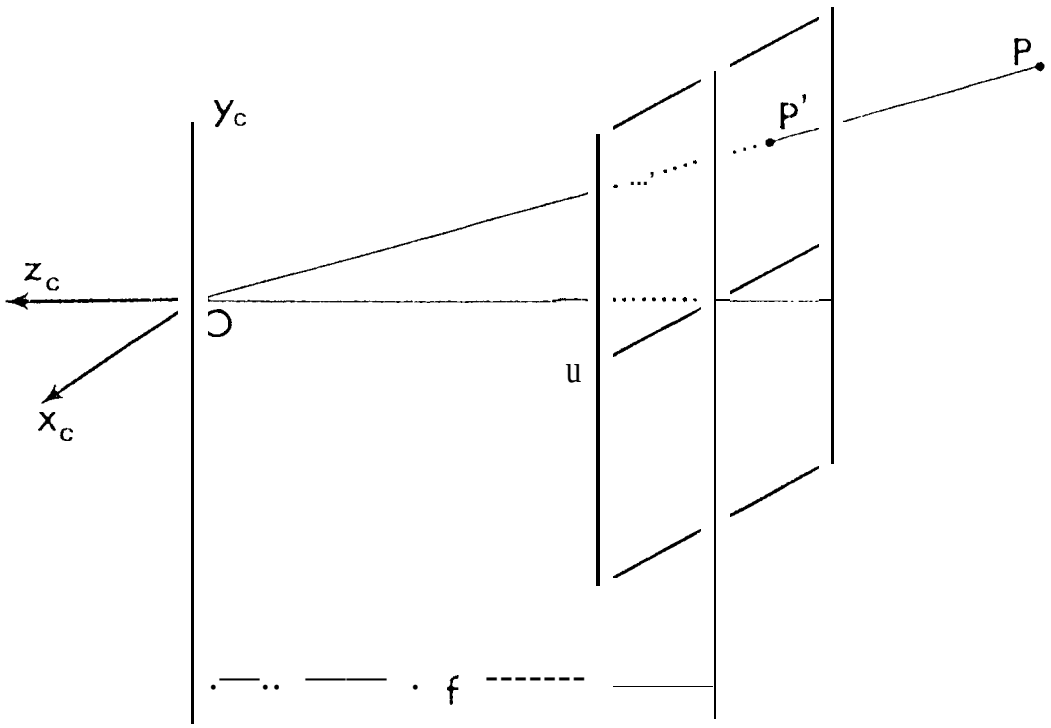
Fig. 5. Calibrated overlays of both robot arm and ORU graphics models on live video images after the camera calibration and object localization using four camera views. (a) side view, (b) oblique view, (c) overhead wide-angle view, and (d) overhead zoom-in view.

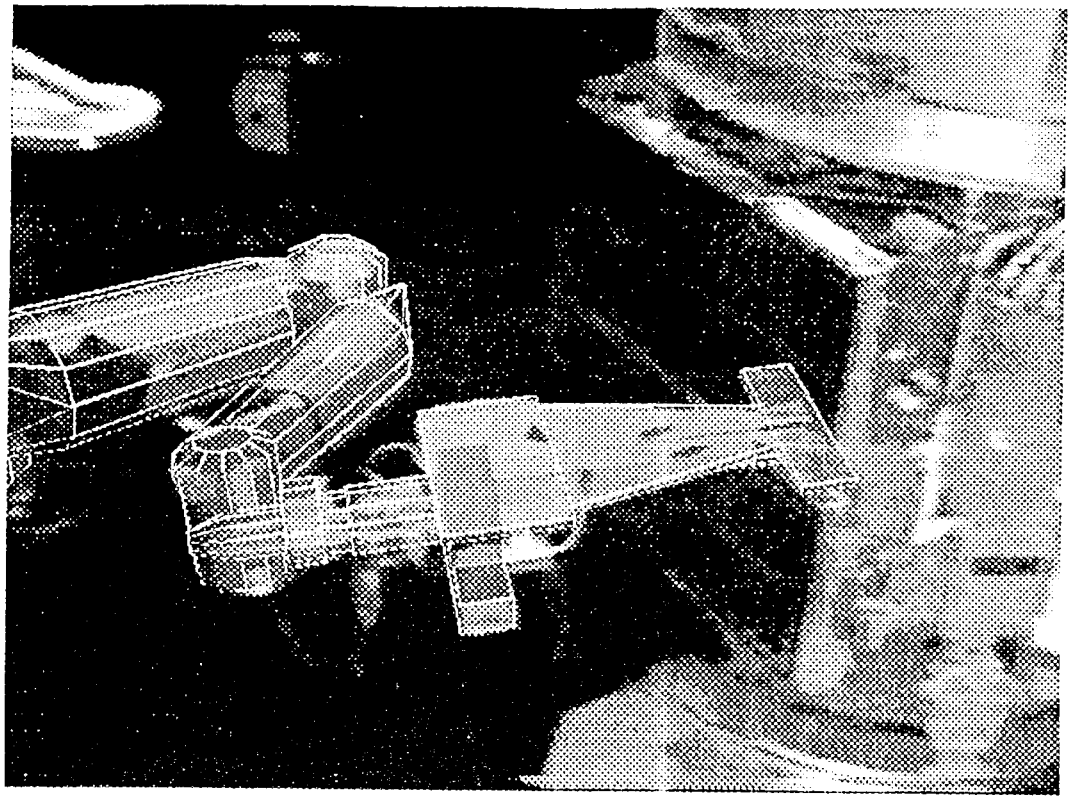
Fig. 6. Snapshots of preview/predictive displays during the JPL/GSFC ORU changeout demonstration. (a) Approach the arm to ORU. (b) Pull out ORU.

Fig. 7. Graphical operator Interface during the task execution.

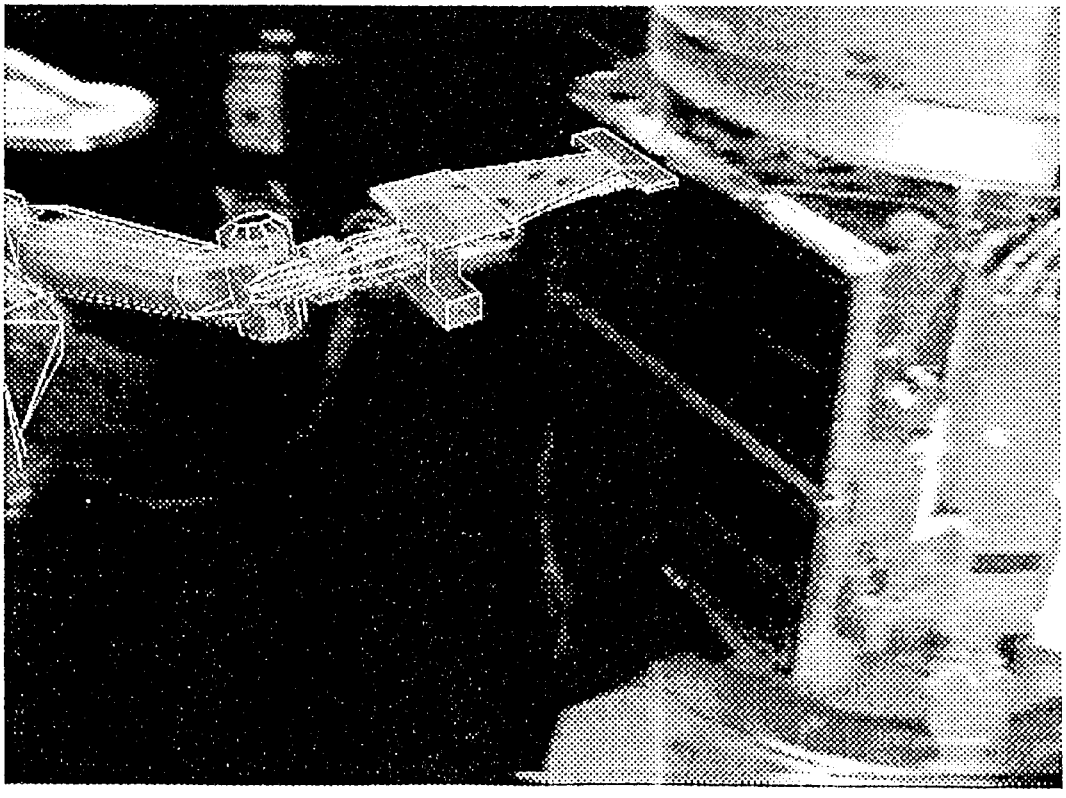

Human with Sense of Tele-Presence
Sensing
Control
Virtual World Model
Sensing
Control
Real World Task Environ-ment
Virtual Environment


Enter Data
Delete & Pause
Clear All
Object:
robot arm
Camera Set Point:
setpoint 2
Robot Set Point:
setpoint 1
n cam X Y Z u v
1 2 -55.179 15.571 138.450 473 260
2 2 -84.596 13.654 136.766 511 191
3 2 -54.707 15.282 130.532 447 257
4 2 -84.124 13.366 128.848 480 185
5 2 -80.376 -7.493 87.183 308 98
6 2 -79.923 -7.770 79.581 275 94
7 2 11.121 9.394 82.262 224 326
8 2 63.418 9.573 142.612 359 402
9 2 35.398 7.938 151.701 402 372
10 2 60.957 9.442 135.067 344 397
11 2 32.937 7.807 144.235 385 367
12 2 20.867 -12.231 103.763 285 289
13 2 18.504 -12.356 96.520 265 283
14 2 26.131 -11.558 94.422 146 274
15 2 39.613 13.229 163.908 430 392
16 2 67.999 14.145 155.831 391 420
17 2 28.383 -5.015 122.361 331 324
18 2 26.292 -4.817 115.036 311 320
Init Blurn
Init Unknown
Save
Cancel
Graphics View
View Control
Rendering
Video Input
prime view
translate
live solid
live video
viewpoints
rotate
live color
off
second vie
zoom
xbitm
image file
swap views
light
geom
tag file
geom video
second vie
Tag Point
Tolerance
Process
select
joint
record cmd
stop
translate
world
playback
rotate
fixed
save traj
create tra
draw view
load traj
trajectory
Creep Object:
Empty Set
traj time:
algorithm
MMS ARM
no scaler
0 sec
grab
pos: 0.5
time scale
release
rot: 0.5
0.33


P
y_c
P'
z_c
O
u
x_c
f

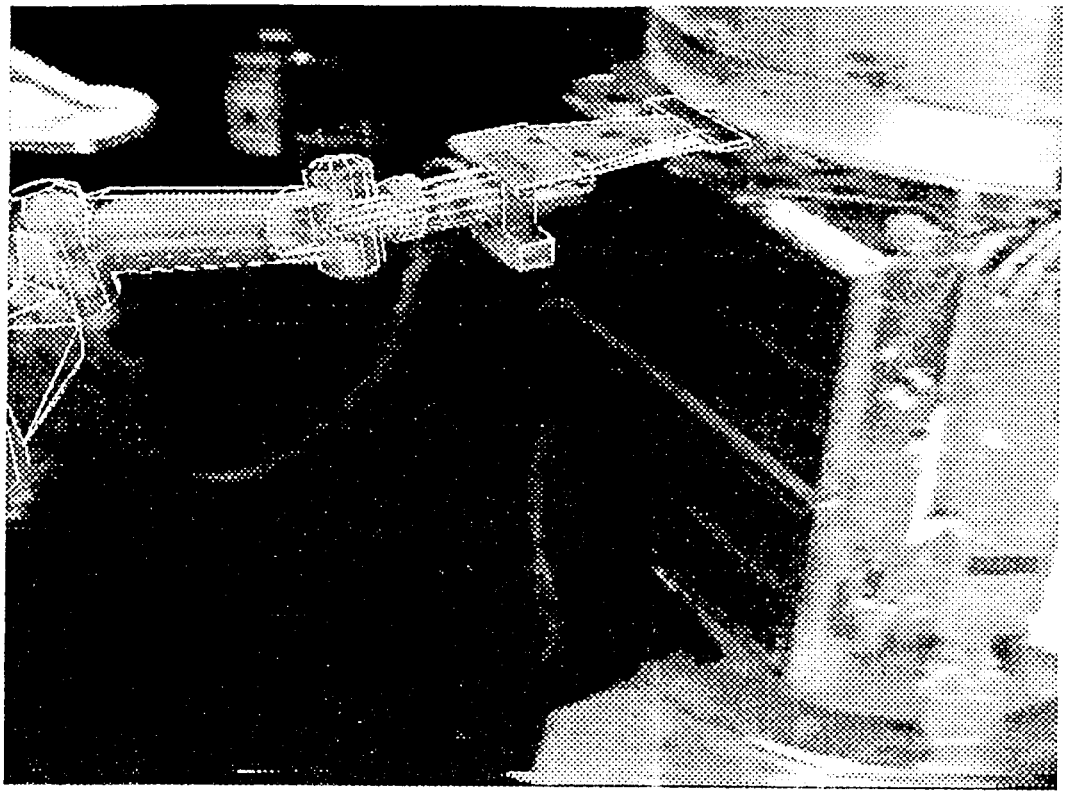


(a)

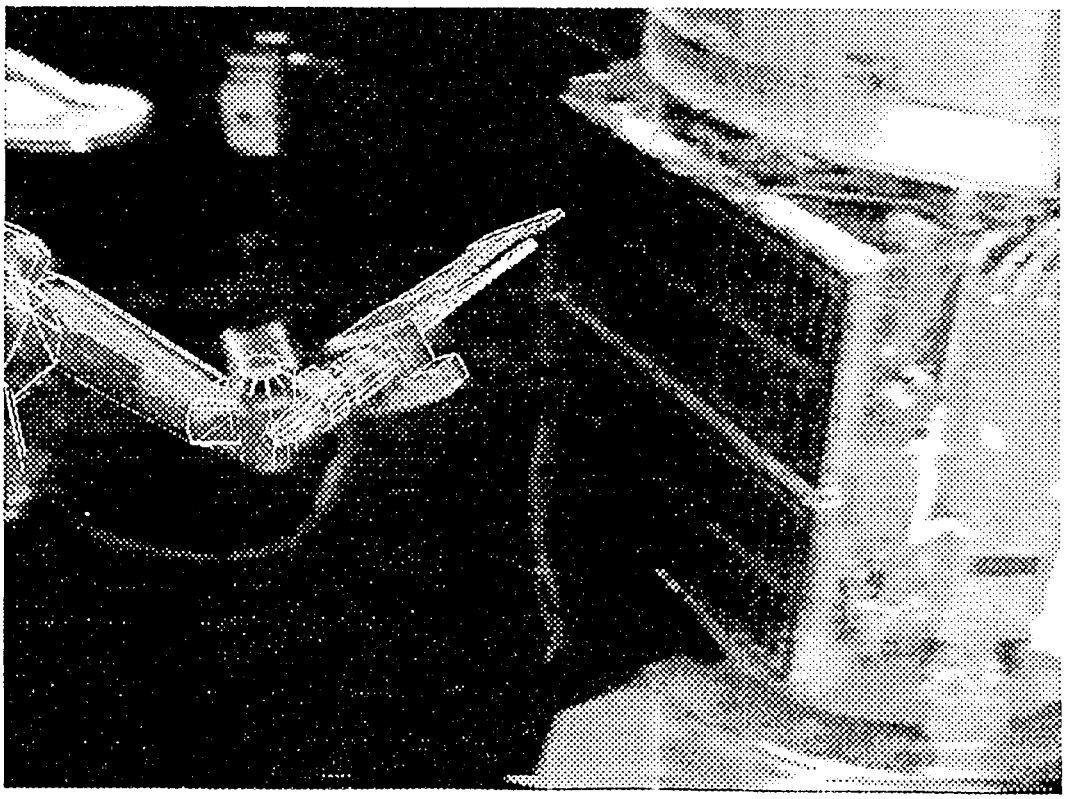


(c)

Fig. 3

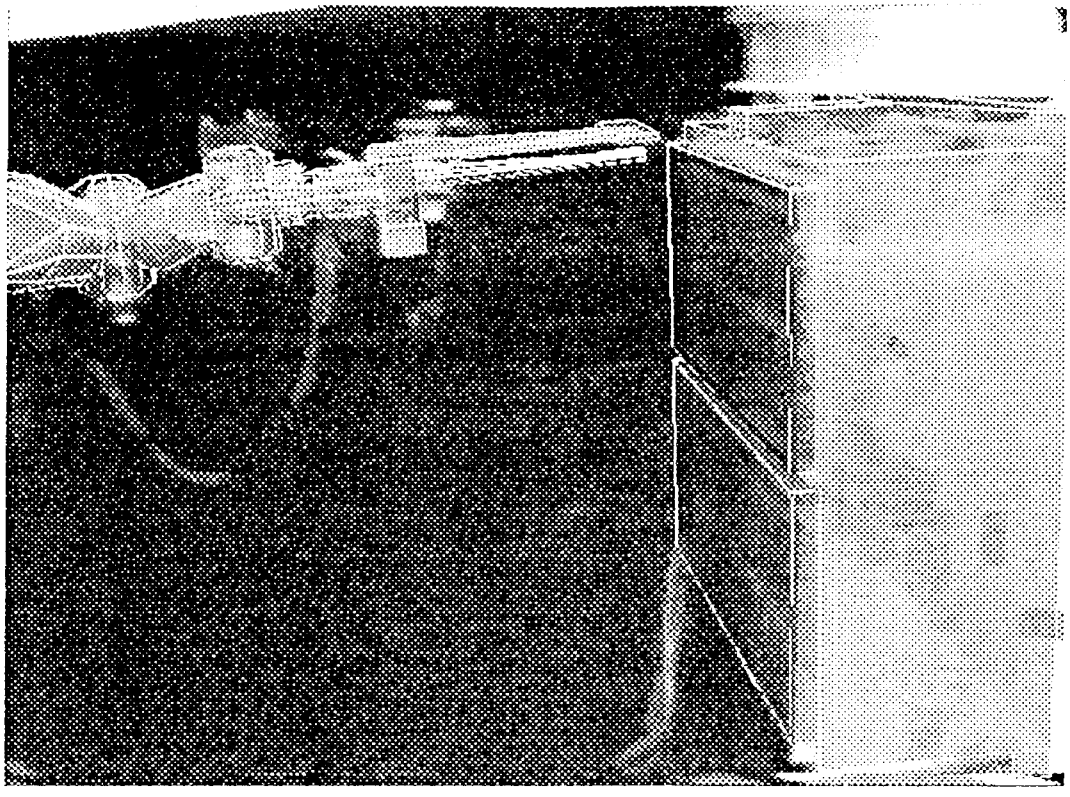


(50)

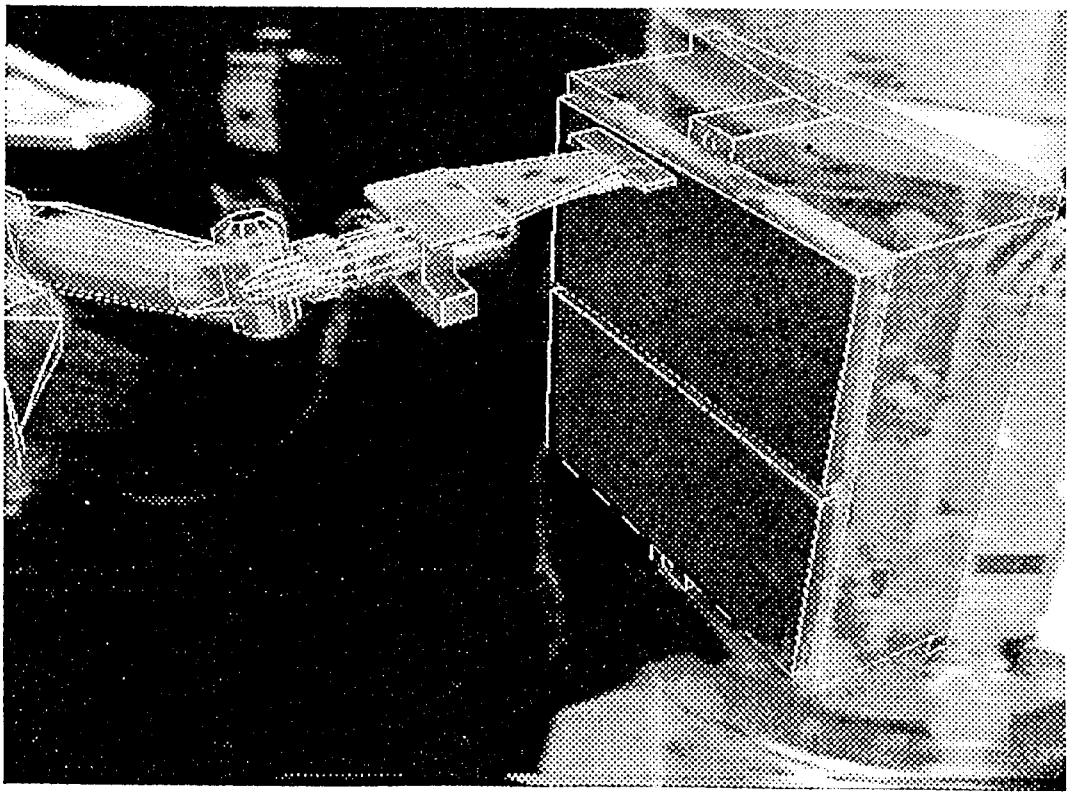


(51)

Fig. 4

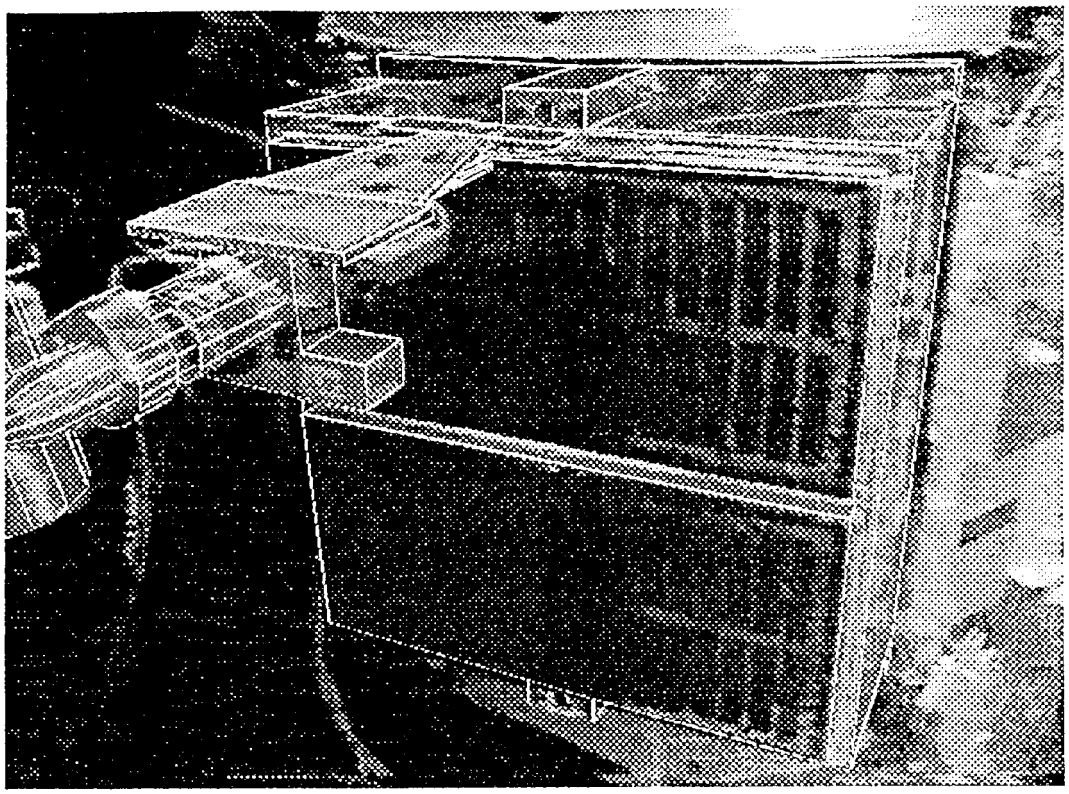


(a)

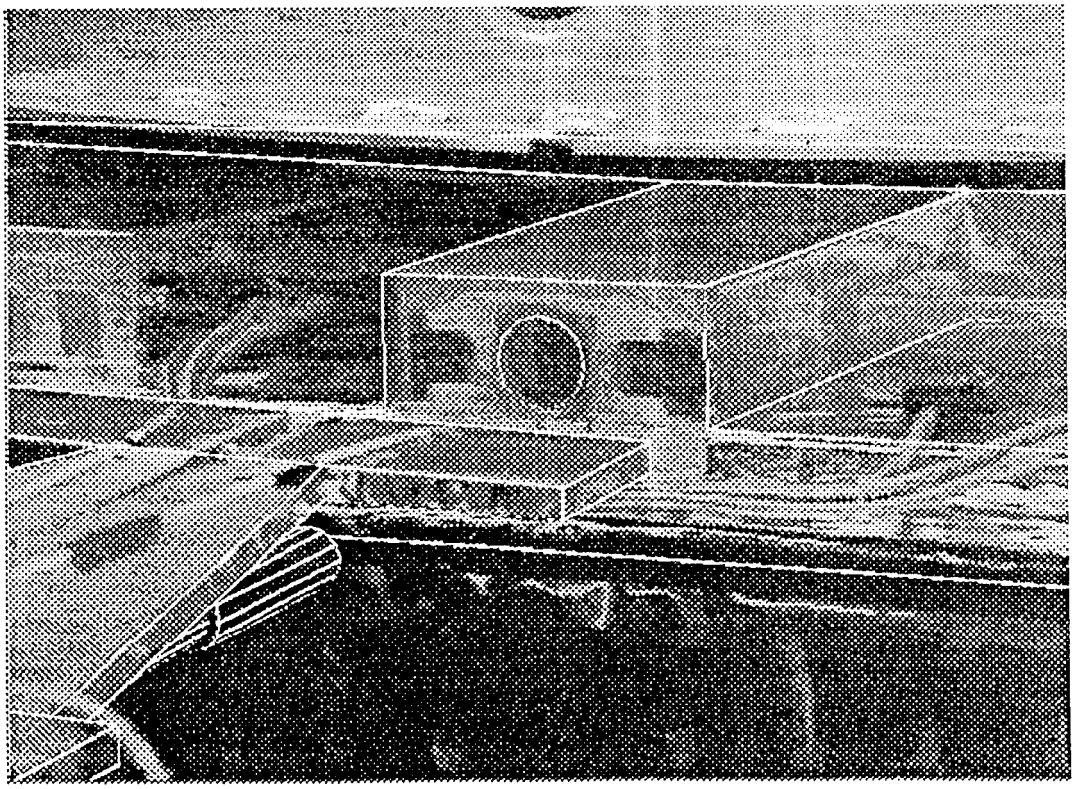


(b)

Fig. 5

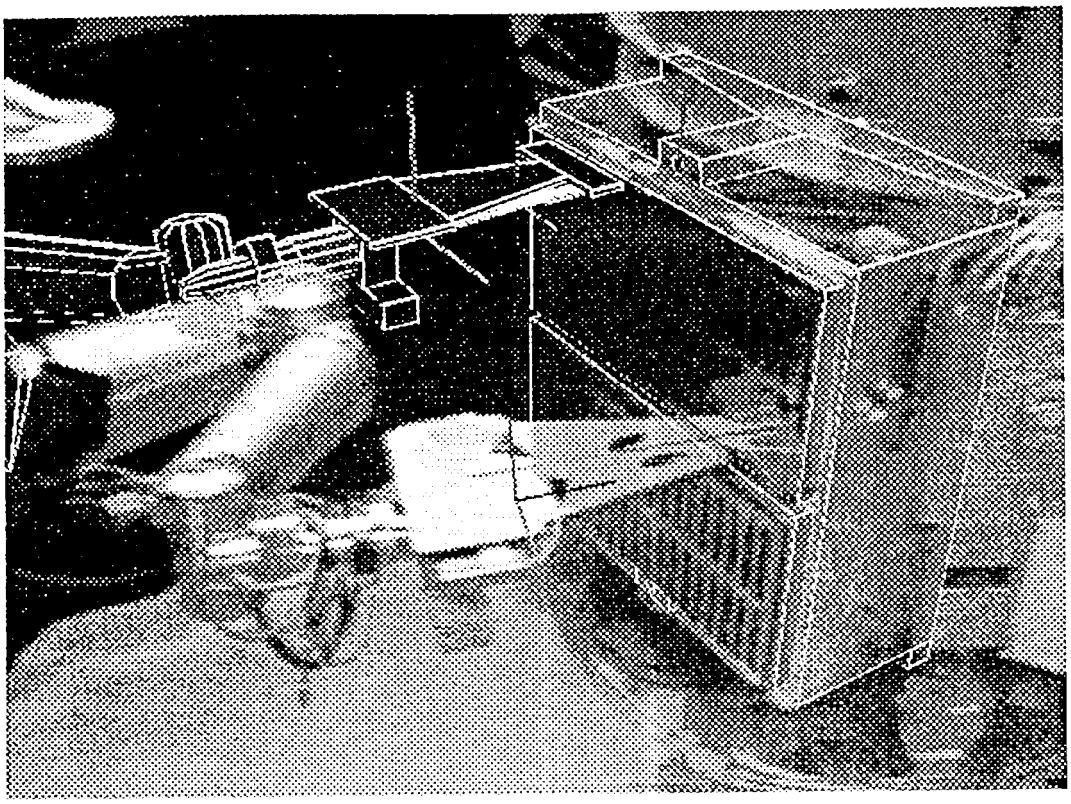


(c)

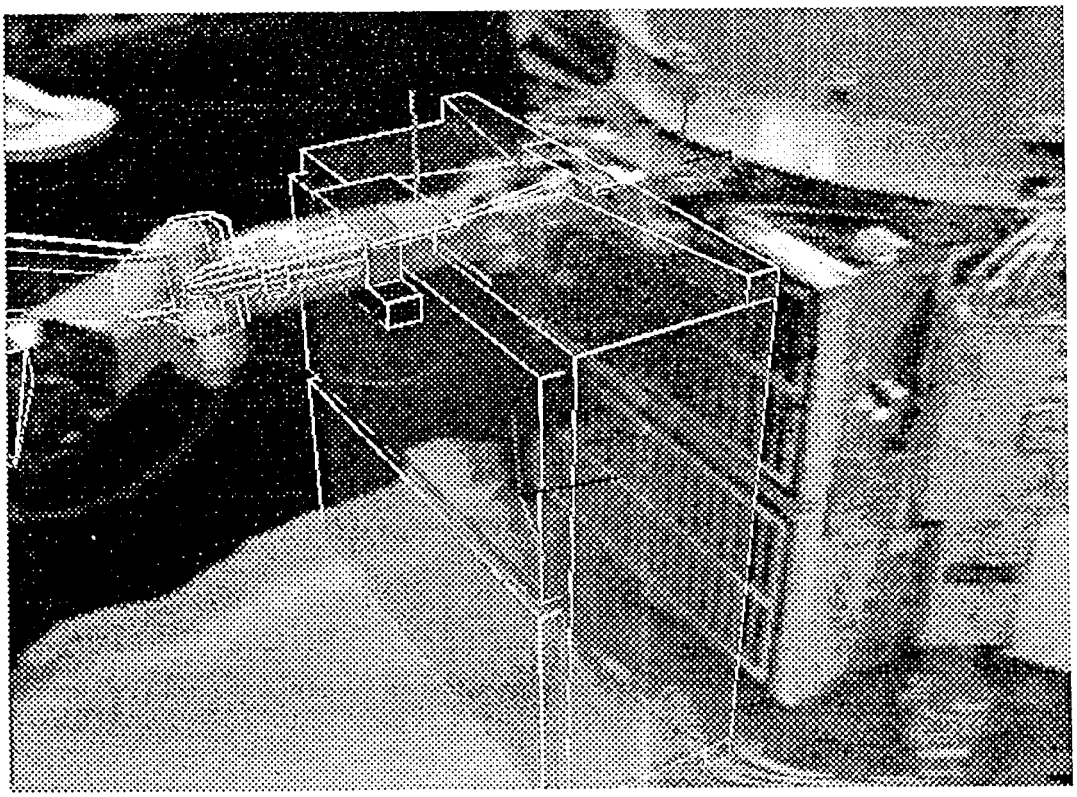


(d)

Fig. 5



(a)



(b)

Fig. 6

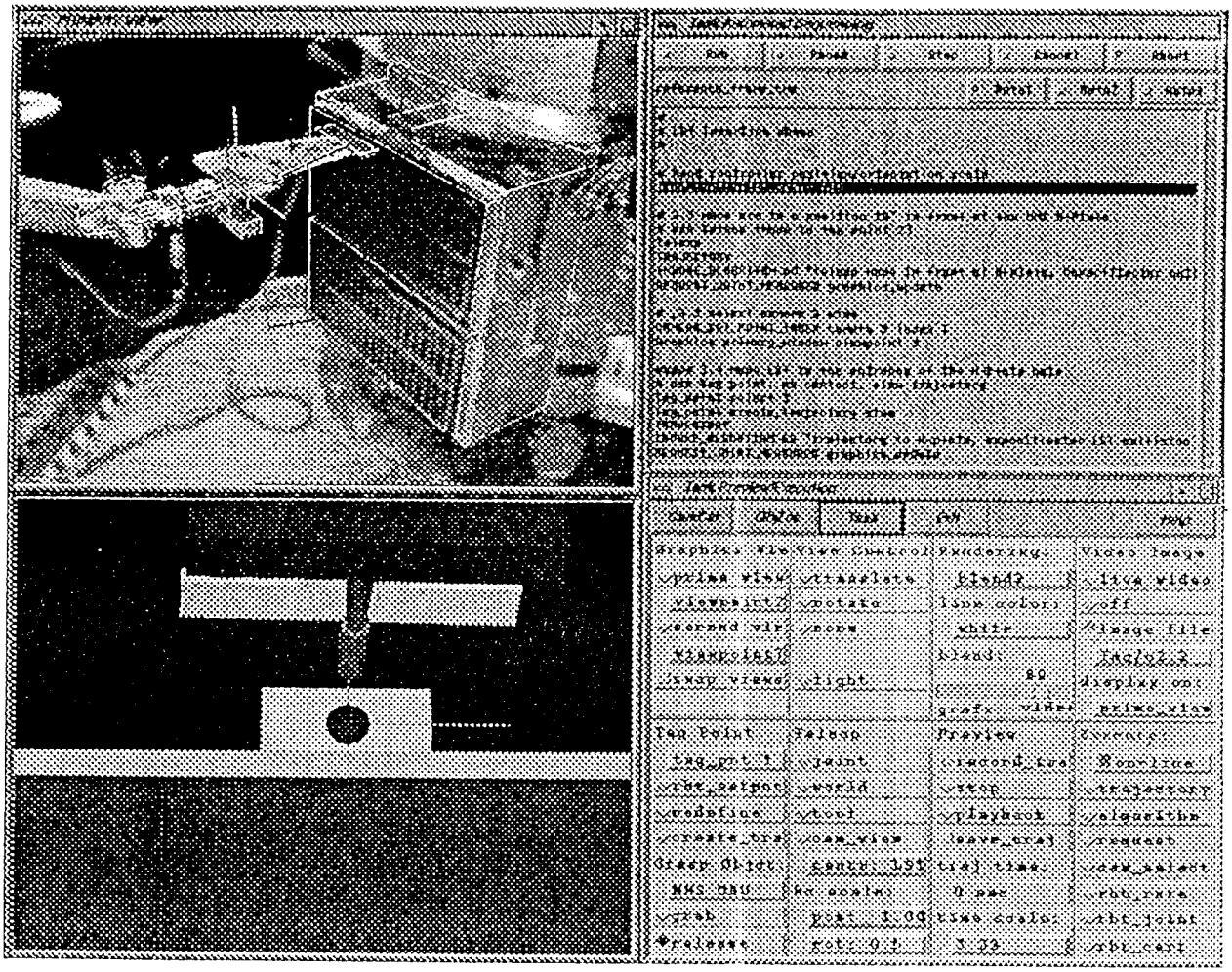


Fig. 7