PERGAMON

Computers & Graphics 25 (2001) 789–798

COMPUTERS & GRAPHICS

www.elsevier.com/locate/cag

Visual and haptic collaborative tele-presence

Adnan Ansar[a,*], Denilson Rodrigues[b], Jaydev P. Desai[c], Kostas Daniilidis[a], Vijay Kumar[a], Mario F.M. Campos[d]

[a] GRASP Laboratory, University of Pennsylvania, Suite 300C, 3401 Walnut Street, Philadelphia, PA 19104, USA
[b] GEAR—Pontifical Catholic University, Belo Horizonte, MG, Brazil, 30535-610
[c] MEM Department, Drexel University, Philadelphia, PA 19104, USA
[d] VER Lab, DCC—Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil, 31270-010

Abstract

The core of a successful sense of presence is a visually, aurally, and haptically compelling experience. In this paper, we introduce the integration of vision and haptics for the purposes of remote collaboration. A remote station acquires a 3D-model of an object of interest which is transmitted to a local station. A user in the local station manipulates a virtual and the remote object as if he/she is haptically and visually at the remote station. This tele-presence feeling is achieved by visually registering the head-mounted display of the local user to the remote world and by dynamically registering the local object both visually and haptically with respect to the remote world. This can be achieved by adequate modeling and feedforward compensation including gravity compensation for the robotic manipulator with which the operator interacts. We present multiple scenarios where such a capability will be useful. One is remote design where a user tests a remotely designed docking station by inserting a virtual laptop into a model of the 3D docking station transmitted from a remote site. Medical robotics provides another possible scenario in which a resident is given surgical training to perform a virtual laparoscopy on a 3D exterior model of a patient, including tomographic registration of anatomical structures. We present results from numerous experiments from both the visual and haptic aspects as well as in integrated form.

Keywords: Augmented reality; Haptics; Tele-presence; Visual registration; Visual tracking

1. Introduction

Approaches to teleoperation and remote control have a long history in automatic control and robotics. Classical teleoperation designs were based on the display of visual information of the remote area in which the actuator pair transfers control from the local to the remote actuator. Since the remote side (and many times also the local side) needs its own control mechanisms and even higher level semi-autonomous behaviors, these efforts belong to the area of telerobotics or remote supervisory control [1]. Telerobotics is widely used now in entertainment, military, space, airborne, underwater, medical applications, and hazardous environments.

The ultimate goal of telerobotics is to accomplish the task at the remote site without necessarily maximizing the sense of "being there" at the remote site. This differs from our approach where the goal is to immerse the user both visually and haptically in the remote environment. Such systems have the following key issues:

- 3D-visual and a force reflecting haptic display.
- Mutual registration of vision and haptics such that the user sees his/her hand but does not see the haptic renderer.
- Tracking of the viewer and displaying visual and haptic stimuli from his/her viewpoint.

*Corresponding author. Tel.: +1-215-898-0355; fax: +1-215-573-2048.

E-mail addresses: ansar@grasp.cis.upenn.edu (A. Ansar), denilsonr@ieee.org (D. Rodrigues), desai@cbis.ece.drexel.edu (J.P. Desai), kostas@grasp.cis.upenn.edu (K. Daniilidis), kumar@grasp.cis.upenn.edu (V. Kumar), kumar@grasp.cis.upenn.edu (V. Kumar), kumar@grasp.cis.upenn.edu (V. Kumar), kumar@grasp.cis.upenn.edu (V. Kumar), kumar@grasp.cis.upenn.edu (V. Kumar), mario@dcc.ufmg.br (M.F.M. Campos).

0097-8493/01/$ - see front matter
PII: S0097-8493(01)00121-2

- 3D-scene acquisition system and recovery of haptic properties of the objects of interest in the remote environment.
- Local controller and predictor to account for delays over the network.

Our requirements are different from the usual haptic renderers where there is no remote site but just a virtual environment [2].

We present a system addressing the above issues based on the following scenario. Consider the schematic of a surgical training system shown in Fig. 1. The 3D exterior model of the patient as well as anatomical details of the operative site acquired by computer tomography (CT), magnetic resonance imaging (MRI), etc. can be transmitted over the internet. The surgeon can then learn the task of operating on a virtual patient in a real environment while receiving real-time visual and haptic feedback on tissue properties from the remote site through tactile sensors in the haptic feedback devices. Another scenario is of a company, say in Taiwan, building a docking station for a laptop newly designed in the United States. The remote site in Taiwan visually acquires a 3D model of the docking station and transmits it to the US site (see Fig. 2).

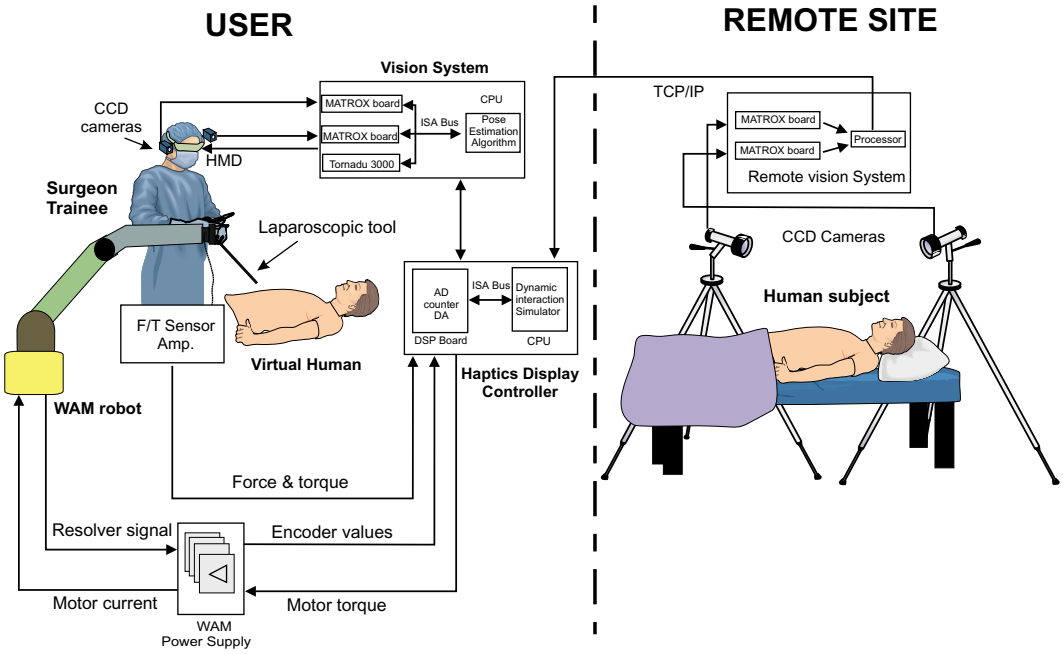
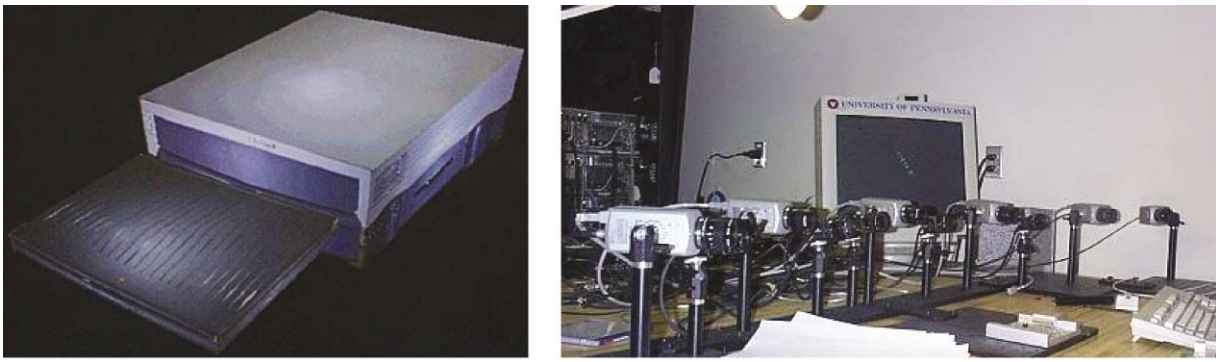


Fig. 1. Tele-presence system for surgical training.



(a) (b)

Fig. 2. (a) The OpenGL representation of the docking station along with the laptop and (b) Camera configuration used for the reconstruction of the docking station.

Fig. 3. User wearing HMD and performing docking procedure.

Only the material properties of the object must be known a priori. At the US site is a manipulator with an end-effector and a user wearing a Head Mounted Display (HMD). The end-effector, with an attached plastic box (as mock-up for the laptop), can reflect forces as the ones felt when someone holds a peg. Cameras mounted on the HMD capture the position of the mock-up and estimate the position of the viewer's head. The user sees and feels in his/her HMD only the laptop and the docking station (see Fig. 3).

We briefly present the main issues and differences to related approaches starting with vision and going into haptics. The most closely related system is the WYSIWYF system at CMU [2]. The main difference between our system and theirs is that the user in our system is immersed into a remote "real" environment instead of a virtual environment. The visual acquisition at the remote site is based on work accomplished in the tele-immersion project [3,4]. The novelty in this approach is the real-time view-point independent scene acquisition and transmission over the network. The 3D reconstruction is model-free and, thus, differs from avatar-based tele-immersion approaches [5].

The local visual registration is based on two novel algorithms for pose estimation. The first uses exactly four points on the vertices of a rectangle. The second accommodates four or more points in arbitrary configuration. A recursive filter is applied for estimation and prediction of targets which could possibly be lost in subsequent frames. We achieve a tracking rate of 17 Hz including the graphics overlay. We emphasize that the approach is purely visual. The head's position in the world is estimated from image data. The challenges of visual registration as opposed to inertial, magnetic, or infrared systems is described in [6]. Real-time approaches using only visual input were presented by [7–9] and are based on pose estimation assuming known calibration. Another approach [10] relaxing the known calibration assumption performs only in a range where the affine projection model holds. Pose estimation is a long studied problem in computer vision. Closed-form solutions exist for three and four points [11–13] and iterative solutions for more than four points [14–19]. Our first algorithm is a closed-form solution for the four vertices of a rectangle of unknown size. The solution is unique, thus avoiding consistency checks, and does not require metric information on the rectangle. There have been recent developments in non-iterative linear solutions for four or more points [20,21]. We have developed an algorithm of this sort [22] which is more robust than the others cited. A unique solution is again obtained for four or more points but in arbitrary configuration.

The operator must receive not only visual but also haptic feedback from the remote environment. There has been significant work done in tactile sensing combined with dextrous manipulation which can be applied to this framework [23]. The master manipulator with which the operator interacts should have near frictionless characteristics and should feel weightless. The system should have a bandwidth of at least 10 Hz [24,25], so that it does not result in unnecessary stress to the operator. Thus, the manipulator should have intrinsically low inertia. The Whole Arm Manipulator (WAM) is ideally suited for such tele-presence tasks.

The goal of a tele-presence system should be to regain the tactile and kinesthetic information that is lost when the operator does not directly manipulate the tools. In the area of minimally invasive surgical training, for example, there are additional limitations in terms of workspace for manipulation and in giving the "feel" of the operative site. One of the main issues in the design of such systems is the incorporation of force-feedback, since the operator loses the feel of the operative site that is so critical for inspection and manipulation of remote objects such as tissues and blood vessels. The incorporation of force-feedback and its benefits have been studied

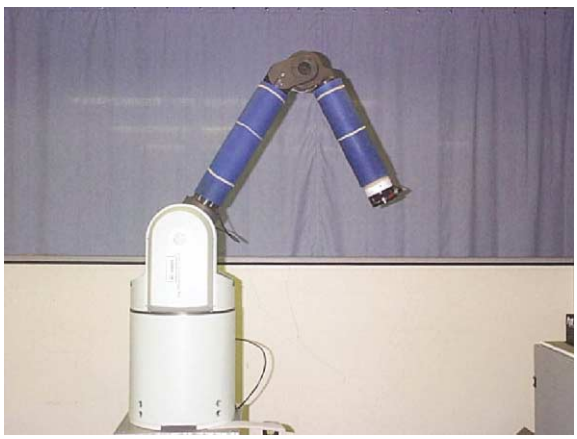


Fig. 4. The whole arm manipulator robot (WAM).

extensively in teleoperation [26–28]. One of the most important issues in such systems is the right balance between fidelity and stability, since they are opposing requirements [29–31]. Time delay is also a critical issue in most teleoperation tasks [32,33] and is more important in tele-presence since a larger amount of information is transferred in an ideal setup.

2. System description

We describe our tele-presence setup. The WAM manufactured by Barrett Technology is a four degree of freedom anthropomorphic robot (see Fig. 4) used as the primary manipulator by the human. The degrees of freedom are: base, shoulder pitch, shoulder roll and elbow. The robot is back driveable, has low friction and intrinsically low impedance. We used the dSPACE DS1103 controller board with the onboard PowerPC processor (PPC 604e) for fast floating point computations at 333 MHz. The board has ADC, DAC and incremental encoder interfaces. An ATI Gamma F/T force-sensor is placed on the end-effector of the WAM.

We capture video with a Sony XC999 color camera mounted onto an HMD and connected to a Matrox Meteor II board. We use an Evans and Southerland Tornado 3000 video card capable of dual VGA output (to two separate monitors or each eyepiece of the HMD). All vision hardware is connected to a Pentium III/550 machine, which does the vision processing. The pose estimation algorithms are written in C. Our rectangular target consists of a piece of cardboard covered in black construction paper. The corners of the target are marked with small squares of colored paper, one of which is a distinct color to allow the pose estimation program to determine orientation.

2.1. Modeling

The structural elements of the WAM are light, and the robot is intrinsically force controllable because it employs a stiff cable transmission and brushless motors. Since the torque transmission from the motor to the joints is via cable transmission, the effect of friction and backlash is reduced. We use Lagrange's formulation to derive the dynamic equations of motion [34]

$$\tau(t) = D(\theta(t))\ddot{\theta}(t) + \mathbf{h}(\theta(t), \dot{\theta}(t)) + \mathbf{c}(\theta(t)). \quad (1)$$

In the above equation, τ is the 4×1 generalized joint torque vector, $D(\theta)$ is the 4×4 inertia tensor, $\mathbf{h}(\theta, \dot{\theta})$ is the 4×1 vector representing the coriolis terms, and $\mathbf{c}(\theta)$ is the 4×1 vector of gravitational forces. We used *Mathematica* to derive the symbolic equations of motion for the WAM. As seen in Eq. (1), viscous and static friction terms are absent. At low speeds and minimum impedance manipulation, the friction forces play a dominant role [35]. In the following section, we will describe the procedure we adopted to estimate the static friction and the modeling we used to incorporate it into the dynamic equations.

2.2. Friction model

Although more elaborate friction models are available [36], we assume a simple model for friction as an initial approach. We model the Coulomb friction and viscous friction as independent of the joint angle. To prevent the stick-slip situation, we define a threshold velocity band of width 2δ centered around the origin where the frictional torque is parabolic with respect to the joint velocity. Given this assumption, the expression for friction torque F_i for the ith joint is

$$F_i(\dot{q}) = \begin{cases} V_i\dot{q}_i + S_i sign(\dot{q}_i), & |\dot{q}_i| > \delta, \\ V_i\dot{q}_i + \frac{S_i}{\delta}\left(\frac{sign(\dot{q}_i)\dot{q}_i^2}{\delta} + 2\dot{q}_i\right), & |\dot{q}_i| \leqslant \delta. \end{cases}$$

2.3. Gravity compensation

In our tele-presence setup, the user interacts with the arm as shown schematically in Fig. 1, changing the position and orientation of the end-effector. Since in an ideal setup, the user should not feel the weight of the manipulator, it is essential to have good gravity compensation. This is achieved by calculating the necessary joint torques required to balance the arm statically in the presence of gravitational forces. Applying the Lagrange formulation for the arm dynamics and making the velocities equal to zero (equilibrium state) results in $\partial U/\partial\theta_i = \tau_i$, where U is the potential energy of the entire system, θ is a 4×1 joint position vector and τ_i is a 4×1 joint torque vector. The potential energy of the system can be written as

$$U = \sum_{i=1}^{4} m_i g^{\mathrm{T}} r_{ci}^0 = \sum_{i=1}^{4} m_i g^{\mathrm{T}} T_i^0 r_{ci}^i. \quad (2)$$

From (2), we can express the joint torques as

$$\tau_i = \frac{\partial U}{\partial\theta_i} = m_4 g^{\mathrm{T}} \frac{\partial T_4^0}{\partial\theta_i} p_{gc4} + m_3 g^{\mathrm{T}} \frac{\partial T_3^0}{\partial\theta_i} p_{gc3},$$

where T_x^0 is the homogeneous transformation from link x to 0, and p_{gc3}, p_{gc4} the centers of mass of links 3 and 4. The set of equations for gravity compensation can be written as $\tau = \psi\mathbf{K}$ where ψ is a 4×4 matrix depending on joint positions and $\mathbf{K}$ is the 4×1 vector of dynamic parameters for gravity compensation. By measuring joint positions and torques for N different static positions, it is possible to write

$$[\tau_{P1}^{\mathrm{T}} \quad \tau_{P2}^{\mathrm{T}} \quad \ldots \quad \tau_N^{\mathrm{T}}]^{\mathrm{T}} = [\psi_{P1}^{\mathrm{T}} \quad \psi_{P2}^{\mathrm{T}} \quad \ldots \quad \psi_N^{\mathrm{T}}]^{\mathrm{T}}\mathbf{K}. \quad (3)$$

Eq. (3) represents an overdetermined system with no exact solution, so we employ a least squares technique

to obtain

$$\hat{\mathbf{K}} = (\psi_{1..N}^{\mathrm{T}}\psi_{1..N})^{-1}\psi_{1..N}^{\mathrm{T}}\tau_{1..N}.$$

2.4. Torque ripple identification

Permanent magnet synchronous motors are used to produce joint torques. In many applications involving low-bandwidth and/or high friction mechanical systems, ripple is unnoticeable at the output [37]. However, in [38] it is shown that the influence of torque ripple affects the force control for an industrial robot. In the high-bandwidth WAM robot, this ripple is felt at the end-effector and we must compensate. The main influence of torque ripple in our setup was in low velocities. To get data to compensate, a PID controller was implemented to spin the motor shaft at very low velocities (0.25 rpm). The torque value sent to the motor was acquired for each of 4096 positions, and compensation is done using the value acquired and feedforward of this value to the motor.

2.5. Visual registration algorithms

There are three relevant coordinate systems: R, the WAM robot coordinate system, F the coordinate system of the fiducial points marked on the object the user holds, and C the coordinate system of the camera. We assume that the camera is calibrated so that image coordinates can be mapped to rays in the C frame. To overlay an object, such as the remote station in the docking scenario, in the R frame, we estimate the transformation from the robot to the camera frame, $T_R^C = T_F^C T_R^F$, where the notation T_A^B refers to the transformation from frame A to B. Because the fiducials' frame is rigidly attached to the end effector, T_R^F is known from the motor readings, so that T_F^C remains to be computed. The mapping T_F^C also enables us to overlay any object in the local user's hands, such as the laptop in the scenario, on the fiducials' frame.

The fiducials, one green and three red, are identified by color segmentation using the Matrox board and the Matrox Image Libraries (MIL Lite). We compute the centroids of the largest clusters of colored points and use these as the image coordinates of the fiducials. Successive estimates for the positions of the fiducials are obtained by tracking them in the image and applying an α–β filter [39] to smooth the results and predict the search area in the next frame. Occluded fiducials are predicted for a maximum of three frames after which the target is declared lost and is visualized at the last found position. The OpenGL libraries are used for texture mapping of the objects we overlay.

2.5.1. Four point pose estimation algorithm

Assume that the four fiducial points on the target are ordered. Let $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ be vectors from the optical center of the camera pointing towards these points. We assume that the $\mathbf{v}_i$ are normalized ($\|\mathbf{v}_i\| = 1$) for all i. Let $\tilde{\mathbf{v}}_i$ be an extension of $\mathbf{v}_i$ to a ray. Any plane P which intersects the optical axis in front of the camera (with positive z-coordinate) will generally intersect the $\tilde{\mathbf{v}}_i$ in four points $\mathbf{w}_i$. The quadrangle $\mathbf{w}_0\mathbf{w}_1\mathbf{w}_2\mathbf{w}_3$ lying in plane P will generally not be a rectangle. We claim that for any orientation of the target rectangle, $\{\mathbf{w}_i\}$ will form the vertices of a rectangle if and only if the plane P is parallel to the plane of the target rectangle. It suffices to recover any such parallel plane to estimate pose. We omit the proof of uniqueness because of space constraints.

Let t_0, t_1, t_2, t_3 be real numbers so that $t_i\mathbf{v}_i$ lie on a plane parallel to that defined by the four fiducials. Since scale is arbitrary at this stage, we assume $t_0 = 1$. This not only sets the global scale, but fixes the specific plane which we recover from the class of planes parallel to the target. Let $\mathbf{s}_i$ be the vector from $t_i\mathbf{v}_i$ to $t_{i+1}\mathbf{v}_{i+1}$ where i is taken mod 4. Then $\mathbf{s}_i = t_{i+1}\mathbf{v}_{i+1} - t_i\mathbf{v}_i$. Since $t_i\mathbf{v}_i$ lie on the vertices of a rectangle, it follows that $\mathbf{s}_{i-1}^{\mathrm{T}}\mathbf{s}_i = 0$ for $i = 0\ldots3$, i.e. the four corners form right angles. After substitution, we have

$$(t_i\mathbf{v}_i - t_{i-1}\mathbf{v}_{i-1})^{\mathrm{T}}(t_{i+1}\mathbf{v}_{i+1} - t_i\mathbf{v}_i) = 0. \tag{4}$$

Any three of the four equations of the form (4) can be combined to obtain a fourth degree polynomial in t_1 alone with coefficients determined by the $\{\mathbf{v}_i^{\mathrm{T}}\mathbf{v}_j\}$. We explicitly solve for t_2 and t_3 as functions of t_1 using *Maple* but omit the very long details here. We now have four independent quartic equations in t_1 from which a linear solution is easily derived. The uniqueness condition on the parallelism class of the supporting plane of the target rectangle guarantees that the four equations will have only one real root in common. If the leading coefficients of all four equations are different from zero, we divide by them to obtain four polynomials of the form

$$E_i = t_1^4 + a_i t_1^3 + b_i t_1^2 + c_i t_1 + d_i, \quad i = 1\ldots4.$$

For $i \neq j$, $E_i - E_j$ is generally a cubic which shares the common real root of the $\{E_i\}$. There will be three such independent polynomials, say F_i, $i = 1\ldots3$. We iterate to obtain two polynomials G_i, $i = 1, 2$ of the form $F_i - F_j$ which are at most quadratic. Finally, $H = G_1 - G_2$ is linear. We solve $H(t_1) = 0$ directly to obtain t_1 and then substitute back to obtain t_2 and t_3.

The coefficients of H depend only on the image coordinates of the fiducial points. If at any step, a leading coefficient becomes zero, we simply have an easier set of equations to solve. Determination of the normal to the plane is now a simple matter of computing $\mathbf{n} = \mathbf{s_i} \times \mathbf{s}_{i+1}$. If the orientation of the target points is known, the cross product can be computed in the correct order to obtain $\mathbf{n}$ in the desired direction. Furthermore, if the size of the target is known, we can uniformly

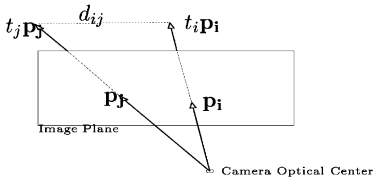


Fig. 5. The fundamental geometric constraint used in the n point algorithm relates the distance between points in the world d_{ij} and the scale factors t_i and t_j associated with the projections $\mathbf{p}_i$ and $\mathbf{p}_j$.

rescale the $\{t_i\}$, so that the $\{\mathbf{s}_i\}$ have the correct lengths. We thus obtain correct global scale and the exact position of the fiducials in frame C. We have recovered T_R^C and can now correctly display an object in the HMD to compensate for the position of the user's head with respect to the world frame R.

2.5.2. N point pose estimation algorithm

The n point algorithm, while not as fast as the closed-form, four point counterpart, is very robust and can be used in more generic situations where more fiducials are needed or in which a rectangular configuration is unsuitable. It requires no initialization and depends on linear algebra techniques rather than iterative methods. Details of this algorithm can be found in [22]. We include only a summary here.

We assume that the coordinates of n points are known in some global frame, and that for every reference point in the world frame, we have a correspondence to a point on the image plane. Our approach is to recover the depths of points by using the geometric rigidity of the target in the form of the $(n(n-1))/2$ distances between n points.

Let $\{\mathbf{w}_i\}$ be n points with projections $\{\mathbf{p}_i\}$. We indicate by d_{ij} the distance between $\mathbf{w}_i$ and $\mathbf{w}_j$. Let $\{t_i\}$ be positive real numbers so that $\mathbf{w}_i = t_i\mathbf{p}_i$. It follows that $d_{ij} = |t_i\mathbf{p}_i - t_j\mathbf{p}_j|$. This is our fundamental constraint (see Fig. 5).

Let $c_{ij} = d_{ij}^2$. Then we have

$$c_{ij} = (t_i\mathbf{p}_i - t_j\mathbf{p}_j)^{\mathrm{T}}(t_i\mathbf{p}_i - t_j\mathbf{p}_j).$$

Letting $p_{ij} = \mathbf{p}_i^{\mathrm{T}}\mathbf{p}_j$, $t_{ij} = t_it_j$ and $\rho = 1$, we rewrite this as

$$t_{ii}p_{ii} + t_{jj}p_{jj} - 2t_{ij}p_{ij} - \rho c_{ij} = 0. \tag{5}$$

This is a homogeneous linear equation in four unknowns. Since $t_{ij} = t_{ji}$, observe that for n points there are $(n(n-1))/2$ equations of the form (5) in $(n(n+1))/2+1$ variables. We write the system as $\mathbf{M}\bar{t} = 0$ where $\bar{t}$ is a vector consisting of the terms $\{t_{ij}\}$ and ρ. Using SVD techniques, this system can be solved up to an $n+1$ dimensional kernel of $\mathbf{M}$, say $\mathbf{K}$, spanned by $n+1$ vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_{n+1}\}$ with $\bar{t} \in \mathbf{K}$. We now use the quadratic constraints imposed by linearizing the original quadratic system to solve for $\bar{t}$. Let $\{\lambda_1, \ldots, \lambda_{n+1}\}$ be the specific values for which

$$\bar{t} = \sum_{i=1}^{n+1} \lambda_i \bar{v}_i. \tag{6}$$

For any integers $\{i,j,k,l\}$ and any permutation $\{i',j',k',l'\}$, we observe that $t_{ij}t_{kl} = t_{i'j'}t_{k'l'}$. Substituting individual rows from the right-hand side of (6) into expressions of this sort leads, after some algebraic manipulation, to homogeneous quadratic constraints on the λ_i. We now linearize the equations in $\lambda_i\lambda_j$ as before to obtain

$$\sum_{i=1}^{n+1}\sum_{j=i}^{n+1} \lambda_{ij} f_{ij}(v_1 \ldots v_{n+1}),$$

where $\lambda_{ij} = \lambda_i\lambda_j$ and f_{ij} are known functions of the vectors $\mathbf{v}_i$ spanning $\mathbf{K}$. We again solve this linear system using SVD. The resulting solution is one-dimensional, but can be restricted to a single point using a length constraint derived from the original system. Having recovered the λ's, we substitute into (6). Now, taking square roots of terms of the form t_{ii}, we recover the coordinates of the world points in the frame of the camera. From the recovered coordinates of the n points in the camera frame, recovery of the world to camera transformation is straightforward [40].

2.6. Remote scene acquisition

A remote scene is reconstructed on-line using the acquisition system extensively described in [4] and pictured in Fig. 2(b). Acquisition is intentionally designed to be independent of the viewpoint of the local user so that both visual and haptic rendering have a refresh rate independent of the acquisition rate. The 3D-acquisition system is based on passive stereo recovery from N views ($N = 7$ in the current system). Depending on the given computational power and plausible visibility constraints, N cameras are combined into triples. A novel multi-baseline algorithm for trinocular stereo [41] is applied to every triple producing a depth-map. All cameras are calibrated with respect to a common world coordinate system. Hence, all depth-maps are also registered with respect to this common reference. Depth-maps continue changing only for regions of the scene which differ from the static background. Such regions can be images of persons or objects like the

Table 1
Percentage of frames tracked for different camera–target distances (in inches)

Distance	Total frames	Percent tracked (%)
15	821	88.4
20	838	87.4
25	873	90.1
30	723	87.6

docking station in the scenario of this paper. The set of points together with registered color texture is sent via TCP/IP to any display station. The current acquisition rate is 2 fps assuming that the dynamic part of the image occupies approximately one quarter of the whole. Acquisition performance was irrelevant in the docking experiment described in this paper because we assumed that the docking station does not move.

3. Results

Using the synchronous grab mode of the Matrox board, we are effectively restricted to a maximum framerate of about 20 Hz. After segmentation, tracking, pose estimation and computation of all frame transformations, we achieve an average of 17 Hz for the vision system. The color segmentation requires 3.5 ms. on average, and subsequent tracking requires 1.5 ms. The four point algorithm requires <100 μs to estimate pose, and all mathematical operations combined require <400 μs. The n point algorithm runs in real-time for up to nine scene points, although we only use it for the four rectangular points in this setting, for which it requires <1 ms. Our first attempt at socket communication has introduced significant delay (40 ms), dropping the framerate to 10 Hz. This delay is entirely in the data transfer, and we hope to eliminate this problem in future implementation.

3.1. Visual registration and tracking

We measure the percentage of frames for which fiducials are successfully tracked versus those for which we must reacquire points. In Table 1, we show data for approximately 800 frames with a randomly moving camera at average distances of 15, 20, 25 and 30 in from the nearest fiducial point. The distances are recovered directly from the pose estimation algorithm.

In Fig. 6, we demonstrate the smoothing effect of the α–β filter by plotting both the measured pixel coordinates of a single fiducial point and the estimates obtained from the filter over 15 frames taken from a camera making a fast, roughly circular motion. In the figure, the corrected trajectory (solid line) is displayed over the trajectory actually recorded by the camera (dotted line).

The visual accuracy of the pose estimation algorithm is checked directly. Four fiducials are placed on the topmost vertices of a cube measuring 7.5 in on all sides and a fifth marker is placed at one of the remaining vertices. We use the corner (three sides of a cube) pictured in Fig. 7 to allow easy placement of the 5th vertex marker. We use our four point algorithm to estimate the location of this extra vertex and compute the distance in pixels between the image of the real marker and its estimated image. The orientation of the camera with respect to the cube is kept roughly constant while the camera is moved in and out from 10 to 30 in (estimated directly from the algorithm) from the closest fiducial point. This is the effective range over which we can acquire data. We plot the error against the distance from the camera to the cube in Fig. 8. Distance is rounded to the nearest inch and pixel errors are averaged for a given distance. We captured data for 1460 frames, with 50–150 frames for each distance. Note that there is no apparent correlation between error and target distance for the range observed.

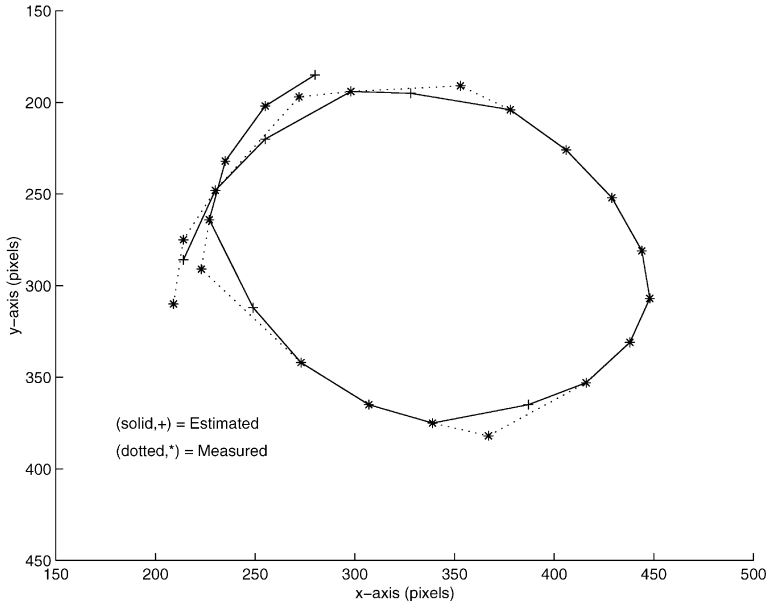


Fig. 6. Smoothing effect of α–β filter. Solid line is trajectory after filter is applied.

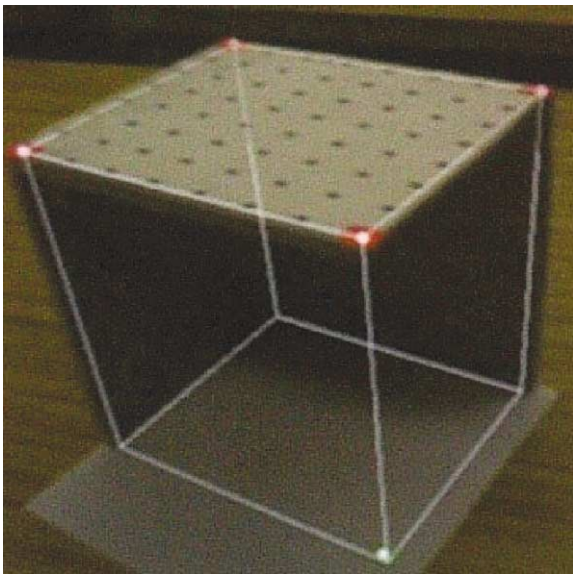


Fig. 7. Arrangement of 4 fiducials and one extra marker on vertices of a cube to evaluate accuracy of pose estimation algorithm. The white lines are superimposed on the image in real time using the pose estimate.

3.2. Haptics

To evaluate the performance of the haptic interface, the user moves the arm by holding a bar attached to the force sensor. Forces are measured at the tip of the WAM robot as the user moves the arm around the docking station and tries to insert the laptop into the station. Fig. 2(a) shows the OpenGL representation of the laptop overlaid on the cardboard target that the user interacts with along with the docking station whose image is acquired from a remote location. Fig. 2(b) shows the actual visual representation of the docking station that is represented in Fig. 2(a).

After visual registration, the image in the head mounted display is used to align the laptop with the docking station and perform the docking maneuver. We introduce a small lateral translation in one eyepiece to simulate a stereo effect. We did not feel any appreciable delay between the visual and haptic systems, and the laptop seemed to touch the station just as we saw contact between the two. Fig. 3 shows the photograph of the user interacting with the WAM robot during an actual docking procedure.

We divide time of interaction with the virtual docking station into the following intervals.

(1) $t_0 < t < t_1$—(*free space*)—user moves laptop in front of docking station without touching it.
(2) $t_1 < t < t_2$—(*front/back*)—laptop in contact with front of docking station.
(3) $t_2 < t < t_3$—(*inside*)—laptop is sliding into docking station.
(4) $t > t_3$—(*inside-bottom*)—laptop touches back of docking station.

From Fig. 9, it is clear that when the user is moving the laptop in free space, the forces change smoothly. In the intermediate stage when the laptop makes contact with any surface of the docking station, a force spike is observed and the user wearing the HMD display "feels" the contact while seeing it. Once contact with the front

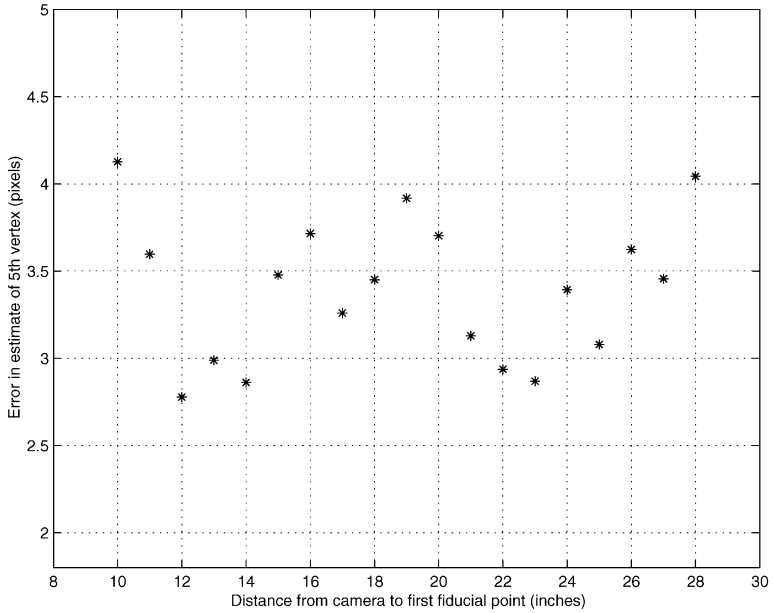


Fig. 8. Pixel error in recovered vertex of cube plotted against distance to camera.

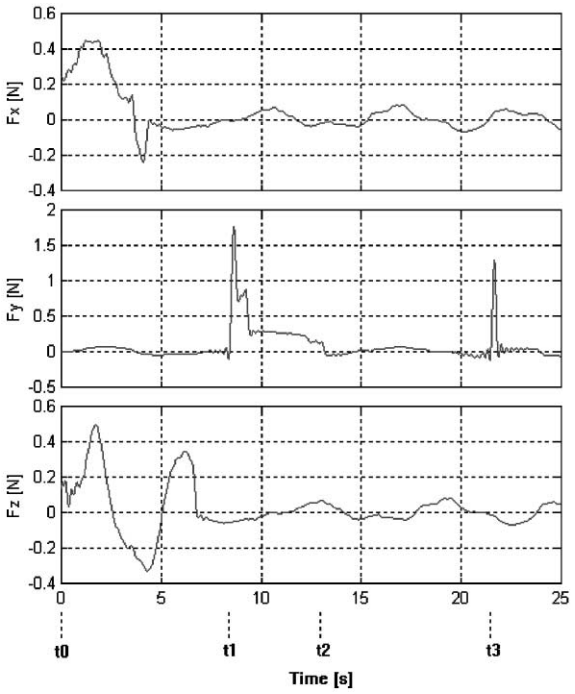


Fig. 9. Forces during the docking procedure.

Table 2
Values of K and B (spring and damping gains) which produce the best subjective results. H, M and L are high, medium and low, respectively

Position	K_x	K_y	K_z	B_x	B_y	B_z
Free space	0	0	0	0	0	0
Top/bottom	0	0	H	0	0	M
Left/right	0	H	0	0	M	0
Front/back	H	0	0	M	0	0
Inside	0	H	H	L	H	H
Inside-bottom	H	H	H	L	H	H

of the docking station is made, the user moves the laptop along its face until it is aligned with the slot. Once the alignment is done, the laptop slides until it reaches the back of the docking station.

The interaction force is computed using a spring-damper model given by $F = K(X).\delta X + B(X).\dot{\delta X}$, where K and B, the spring and damping gain values, respectively, are constant for specific ranges of the position vector X in the Cartesian space of the task. Table 2 depicts the values of K and B that produced the best results during the docking procedure. By "best" we mean the subjectively overall better sensation perceived by several users who performed the procedure in the lab.

H, M and L stand for high, medium and low gains, respectively. In free space, K and B are zero, and we only need to compensate for forces due to friction and gravity to make the arm "weightless". At those positions, where a collision is implied, different values of K and B are used according to Table 2. This will provide different resulting forces being fed back to the user. Since a 4-DOF arm was used, it is evident that it is unable to respond to general forces and torques to the user's hand (6-DOF). This will be addressed with the installation of an adequately designed 2-DOF wrist joint at the end of the last link. With the present configuration, our system is being used to evaluate the synchronization of the pose estimation performed by the vision system module and the haptic display controller for the docking procedure already described.

4. Discussion

In this paper, we have accomplished the first stage of the tele-presence setup, namely, we are able to track and estimate pose in real time and render in an OpenGL environment. The depth recovery is sufficiently accurate to match the data obtained from the force interaction of the local object with the remote environment. In order to accomplish this we have a reasonably good model of the robot dynamics, including gravity compensation, friction modeling and torque ripple compensation. We have demonstrated the validity of our approach with the example of docking a laptop.

There are several challenging issues that still need to be addressed. These include visual representation and rendering in real time of an object in a remote environment and transferring the haptic information via the dynamic simulator as shown in Fig. 1.

Acknowledgements

This work has been supported by NSF IIS-0083209, ARO/MURI DAAH04-96-1-0007, NSF CDS-97-03220, DARPA-ITO-DABT63-99-1-0017, Penn Research Foundation, and Advanced Network and Services.

References

[1] Sheridan TB. Telerobotics, Automation, and Human Supervisory Control. Cambridge, MA: MIT Press, 1992.
[2] Yokokohji Y, Hollis R, Kanade T. Wysiwyf display: a visual/haptic interface to virtual environment. Presence 1999;8(4):412–34.
[3] Raskar R, Welch G, Cutts M, Lake A, Stesin L, Fuchs H. The office of the future: a unified approach to image-based. ACM SIGGRAPH 1998;179–88.
[4] Mulligan J, Daniilidis K. View-independent scene acquisition for tele-presence. Proceedings of the International Symposium on Augmented Reality, Munich, Germany, October 5–6, 2000, pp. 105–10.
[5] Leigh J, another 42 authors. A review of tele-immersion applications in the CAVE research network. Proceedings of the IEEE Virtual Reality Annual International Symposium, 1999.

[6] Azuma RT. A survey of augmented reality. Presence 1997;7:355–85.
[7] Koller D, Klinker G, Rose E, Whitaker R, Tuceryan M. Automated camera calibration and 3D egomotion estimation for augmented reality applications, International Conference on Computer Analysis of Images and Patterns, 1997. p. 199–206.
[8] Bajura M, Neumann U. Dynamic registration correction in video-based augmented reality systems. IEEE Computer Graphics and Application 1995;15(5):52–60.
[9] Mellor JP. Enhanced reality visualization in a surgical environment. Massachusetts Institute of Technology Artificial Intelligence Laboratory, vol. 1544, 1995.
[10] Kutulakos K, Vallino J. Calibration-free augmented reality. IEEE Transactions on Visualization and Computer Graphics 1998;4(1):1–20.
[11] Haralick RM, Lee C, Ottenberg K, Nolle M. Analysis and solutions of the three point perspective pose estimation problem. CVPR 1994. p. 592–8.
[12] Fischler M, Bolles RC. Random sample consensus: a paradigm for model fitting and automatic cartography. Communications of the Association for Computing Machinery 1981;6:381–95.
[13] Horaud R, Canio B, Leboulleux O. An analytic solution for the perspective 4-point problem. Computer Vision, Graphics, and Image Processing 1989;1:33–44.
[14] Lu C-P, Hager G, Mjolsness EM. Fast and globally conergent pose estimation from video images. IEEE Trans. Pattern Analysis and Machine Intelligence 2000;22:610–22.
[15] DeMenthon D, Davis LS. Inverse perspective of a triangle. ECCV90, 1990. p. 369–73.
[16] Liu Y, Huang TS, Faugeras OD. Determination of camera location from 2-D to 3-D line and point correspondences. PAMI 1990;12:28–37.
[17] Kumar R, Hanson AR. Robust methods for estimating pose and a sensitivity analysis. IU 1994;60:313–42.
[18] Holt RJ, Netravali AN. Camera calibration problem: some new results. IU 1991;54:368–83.
[19] Lowe DG. Three-dimensional object recognition from single two-dimensional images. Artificial Intelligence 1987;31:355–95.
[20] Quan L, Lan Z. Linear *n*-point camera pose determination. IEEE Transactions on Pattern Analysis and Machine Intelligence 1999;21:774–80.
[21] Fiore PD. Efficient linear solution of exterior orientation. IEEE Transactions on Pattern Analysis and Machine Intelligence 2001;23:140–8.
[22] Ansar A, Daniilidis K. Linear augmented reality registration. International Conference on Computer Analysis of Images and Patterns, Warsaw, Poland, September 5–7, 2001, to appear.
[23] Howe RD. A force-reflecting teleoperated hand system for the study of tactile sensing in precision manipulation. Proceedings of the IEEE International Conference on Robotics and Automation, 1992. p. 1321–6.
[24] Brook W, Mannema DP. Master-slave manipulator performance for various dynamic characteristics and positioning task parameters. IEEE Transactions on Systems Man and Cybernetics 1990;10:764–71.
[25] Uebel M, Ali M, Minis I. The effect of bandwidth on telerobot system performance. IEEE Transactions on Systems Man and Cybernetics 1994;24(2):342–8.
[26] Das H, Zak H, Kim WS, Bejczy AK, Schenker PS. Operator performance with alternative manual control modes in teleoperation. Presence 1992;1(2):201–18.
[27] Hannaford B, Wood L, McAffee DA, Zak H. Performance evaluation of a six-degree generalized force-reflecting teleoperator. IEEE Transactions on Systems Man and Cybernetics 1991;21(3):620–33.
[28] Kim WS, Hannaford B, Bejczy AK. Force-reflection and shared compliant control in operating telemanipulators with time delay. IEEE Transactions on Robotics and Automation 1992;8:176–85.
[29] Hannaford B. A design framework for teleoperators with kinesthetic feedback. IEEE Transactions on Robotics and Automation 1989;5:426–34.
[30] Raju GJ, Verghese G, Sheridan TB. Design issues in 2-port network models of bilateral remote manipulation. Proceedings of the IEEE International Conference on Robotics and Automation, 1989. p. 1316–21.
[31] Sheridan TB. Human factors in telesurgery. In: Computer integrated surgery. Cambridge, MA: MIT Press, 1996. p. 223–9.
[32] Penin LF, Matsumoto K, Wakabayashi S. Force reflection for time-delayed teleoperation of space robots. IEEE International Conference on Robotics and Automation 2000;4:3120–5.
[33] Stein MR, Paul RP. Operator interaction, for time-delayed teleoperation, with a behavior-based controller. IEEE International Conference on Robotics and Automation 1994;1:231–6.
[34] An CH, Atkeson CT, Hollerbach JM. Model-based control of a robot manipulator. Cambridge, MA: MIT Press, 1988.
[35] Desai JP, Howe RD. Towards the development of a humanoid arm by minimizing interaction forces through minimum impedance control. International Conference on Robotics and Automation, Seoul, Korea, May 21–26, 2001, pp. 4214–9.
[36] Armastrong-Helouvry B, Dupont P, De Wit CC. A survey of models, analysis tools and compensation methods for the control of machines with friction. Automatica 1994;30:1083–138.
[37] Colamartino F, Marchand C, Razek A. Torque ripple minimization in permanent magnet synchronous servodrive. IEEE Transactions on Energy Conversion 1999;14(3):616–21.
[38] Ferreti G, Magnani G, Rocco P. Force oscillations in contact motion of industrial robots: an experimental investigation. IEEE/ASME Transactions on Mechatronics 1999;4(1):86–91.
[39] Bar-Shalom Y, Fortmann T. Tracking and data association. Orlando: Academic Press, 1988.
[40] Horn BKP, Hilden HM, Negahdaripour S. Closed-form solution of absolute orientation using orthonormal matrices. Journal of the Optical Society of America A 1988;A5:1127–35.
[41] Mulligan J, Daniilidis K. Trinocular stereo for non-parallel configurations. Proceedings of the International Conference on Pattern Recognition, Barcelona, Spain, September 1–3, 2000, pp. 567–70.