Proceedings of the 1997 IEEE
International Conference on Robotics and Automation
Albuquerque, New Mexico - April 1997

Integrating Force and Vision Feedback Within Virtual Environments for Telerobotic Systems

Bradley J. Nelson

Department of Mechanical Engineering
University of Illinois at Chicago
842 West Taylor Street
Chicago, Illinois 60607-7022

Pradeep K. Khosla

Dept. of Electrical and Computer Eng.
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, Pennsylvania 15213-3890

Abstract

Traditional telerobotic systems often provide feedback to the user through a variety of sensing modalities, for example through live video imagery, force reflection, or acoustic signals. When a supervisor guides a robotic task while immersed in a virtual environment, geometric representations of the world are provided as feedback. In this case, the supervisor interacts with the environment by visually observing these virtual objects and directing their motion. At issue is how to appropriately use various sensing modalities provided by disparate sensors in a system of this configuration. This paper focuses on assimilating the disparate feedback provided by force and vision sensors for telerobotic systems guided from within virtual environments. A framework for feedback assimilation is described based on the concept of sensor resolvability. Sensor resolvability is used in two ways, to update the virtual environment and to guide the desired task in the real world. Resolvability selects the appropriate sensing modality to use in updating the virtual environment within which the supervisor is immersed. It is also used to direct semi-autonomous agents that interact directly with the real world to perform the desired task. Experimental results demonstrate the significant advantages of assimilating disparate sensory feedback throughout a telerobotic task using the concept of sensor resolvability.

1. Introduction

When a supervisor guides a telerobotic system from within a virtual environment, geometric representations of the world in the form of CAD models provide feedback to the user. The supervisor interacts with the environment by visually observing these virtual objects and directing their motion. However, traditional telerobotic systems often provide feedback to the user through a variety of sensing modalities, for example through live video imagery, force reflection, or acoustic signals. There is an obvious need to use feedback from widely differing sensors in order to properly monitor various aspects of the task, particularly for complex tasks. For teleoperated object manipulation, force and vision feedback are the most important sensing modes due in part to their complementary nature. Vision is useful for aligning objects; force ensures reasonable contact forces are maintained as parts mating occurs.

For telerobotic manipulation from within virtual environments, there is an inherent problem in integrating feedback from disparate sensors such as force and vision. This is because the virtual environment rather than the human operator must integrate the feedback, so the virtual environment must have the capability of integrating feedback from various sensors and sensing modalities. However, conventional techniques for sensor integration operate in some common space closely related to the particular sensors used in the system, often using a probabilistic weighting method for combining information from different sensors [3][14][16]. This has obvious drawbacks for integrating disparate feedback such as force and vision, since the two sensor spaces are quite different. Conventional sensor integration also assumes that a temporally accurate cross-coupling between sensors can be modeled in sensor space. Disparate sensing modes are often appropriate during different stages of the task, making a temporal comparison of the data sets meaningless during most of the task.

In this paper, we describe and experimentally verify a framework for telerobotic manipulation within virtual environments using disparate sensing modalities. Within the framework, the real world is represented by dynamically updated 3D geometric models of objects in the environment. The supervisor directs the task by directing the relative motion of these object models. Each model is augmented by sensor mappings representing the vision and force sensors used to provide feedback during task execution. These augmented environment models are referred to as *object schemas*, because they simultaneously represent the task and provide an understanding of how the environment is perceived, similar to the more general concept of *schema* used by cognitive psychologists [12]. The mappings are used in two ways. First, they predict how the sensors will view the environment model while the given task is being executed. This provides reference inputs to

0-7803-3612-7-4/97 $5.00 © 1997 IEEE

autonomous sensor-based controllers referred to as *port-based agents* which then guide task execution. Second, object schemas use their sensor mappings to determine the appropriate sensing modality to be used in updating the virtual environment. Experimental results demonstrate the significant advantages of assimilating disparate sensory feedback throughout a telerobotic task using the concept of sensor resolvability.

2. A Virtual Environment of Object Schemas

A key component of our system framework is the structure and content of the virtual environment task representation. In order to provide the supervisor with the capability to reason about object manipulation, the internal representation must contain three dimensional geometric knowledge of the objects being manipulated. The internal representation must also be capable of being correlated with vision and force information in order to direct the visually servoed/force controlled manipulator that performs the task. This implies that a sensor-based representation of the task must also exist internally.

2.1. Object Schemas

The task representation is defined in terms of objects. Each object representation we call an *object schema.* Our object schema definition includes a geometric environment model of an object augmented by sensor mappings that describe how the object will be perceived by the actual system sensors. Forward projective sensor mappings and the associated Jacobian matrix for each sensor that exists in the system provide this representation.

Figure 1 shows a diagram of an object schema, including the current internal pose of the geometric model; its desired pose (for controllable schemas) as determined by a supervisor; and sensor mappings used to direct variable sensor parameters as well as update the current internal pose of the schema based on actual sensor feedback. Figure 2 shows sensor mappings for vision and force sensors. The pseudoinverse of the Jacobian mapping used for determining sensor resolvability is also used to "servo" the geometric model in order to reduce errors between the current internal visual representation of the object and the actual visual representation.

The concept of *sensor resolvability,* which we proposed

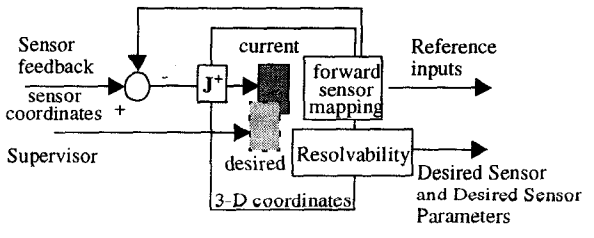

Figure 1. An object schema diagram.

in [11], provides a measure of the ability of both force and vision sensors to resolve positions and orientations in task space. This provides a technique for assimilating the data from the two sensors. Resolvability is a function of the Jacobian of the mapping from task space to sensor space. We desire a matrix form of the Jacobian which contains both intrinsic and extrinsic sensor parameters in order to analyze the effects of these parameters on the structure of the Jacobian. For any sensor system, we desire an equation of the form

$$\delta \mathbf{x}_S = \mathbf{J}(\phi)\delta \mathbf{X}_T \tag{1}$$

where $\delta \mathbf{x}_S$ is an infinitesimal displacement vector in sensor space, $\mathbf{J}(\phi)$ is the Jacobian matrix and is a function of the extrinsic and intrinsic parameters of the sensor (as well as the type and number of features being tracked for vision sensors), and $\delta \mathbf{X}_T$ is an infinitesimal displacement vector in task space.

By performing a singular value decomposition [8] on the task space to sensor space Jacobian, and analyzing the singular values of $\mathbf{J}$ and the eigenvectors of $\mathbf{J}^T\mathbf{J}$ which result from the decomposition, the directional properties of the ability of the sensor to resolve positions and orientations becomes apparent. These directional properties can be represented by the resolvability ellipsoid. The following sections briefly describe the derivation of the Jacobian mapping and analyze the Jacobian for various vision and force sensor configurations.

2.2. Vision Resolvability

For a single camera, forward sensor mappings are simply perspective projection mappings and are of the form

$$x_S = \frac{fX_C}{s_x Z_C} \qquad y_S = \frac{fY_C}{s_y Z_C} \tag{2}$$

where x_S and y_S are the projected image coordinates of a point on the object in the internal representation located with respect to the camera frame at (X_C,Y_C,Z_C), f is the focal length of the camera lens, and s_x and s_y are the pixel dimensions of the CCD array. For the experimental results to be presented, an orthogonal stereo pair is used. Figure 2

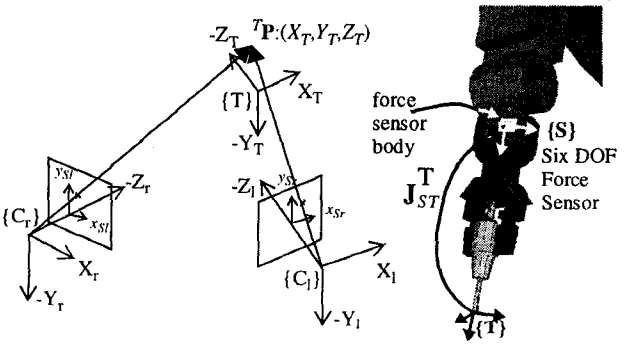

Figure 2. Task frame-sensor frame definitions for a stereo camera system and a manipulator force sensor.

shows the coordinate frame definitions for this type of camera-lens configuration. If the axes are aligned as shown in the figure, the Jacobian mapping from task space to sensor space for a single feature can be written as

$$\mathbf{J}_v = \begin{bmatrix} \frac{f}{s_x Z_{Cl}} & 0 & -\frac{x_{Sl}}{Z_{Cl}} & -\frac{x_{Sl}Y_T}{Z_{Cl}} & \left[\frac{fZ_T}{s_x Z_{Cl}} + \frac{x_{Sl}X_T}{Z_{Cl}}\right] & -\frac{fY_T}{s_x Z_{Cl}} \\ 0 & \frac{f}{s_y Z_{Cl}} & -\frac{y_{Sl}}{Z_{Cl}} & -\left[\frac{fZ_T}{s_y Z_{Cl}} + \frac{y_{Sl}Y_T}{Z_{Cl}}\right] & \frac{y_{Sl}X_T}{Z_{Cl}} & \frac{fX_T}{s_y Z_{Cl}} \\ \frac{x_{Sr}}{Z_{Cr}} & 0 & \frac{f}{s_x Z_{Cr}} & \frac{fY_T}{s_x Z_{Cr}} & \frac{x_{Sr}Z_T}{Z_{Cr}} - \frac{fX_T}{s_x Z_{Cr}} & \frac{x_{Sr}Y_T}{Z_{Cr}} \\ \frac{y_{Sr}}{Z_{Cr}} & \frac{f}{s_y Z_{Cr}} & 0 & -\frac{fZ_T}{s_y Z_{Cr}} & \frac{y_{Sr}Z_T}{Z_{Cr}} & \frac{y_{Sr}Y_T}{Z_{Cr}} - \frac{fX_T}{s_y Z_{Cr}} \end{bmatrix} \quad (3)$$

In (3), we assume the camera-lens parameters are identical for both cameras. The other terms in (3) correspond to Figure 2.

2.3. Force Resolvability

For force sensor resolvability, the Jacobian mapping of the force sensor is represented by

$$\delta \mathbf{x}_S = \mathbf{J}_f(t)\delta \mathbf{X}_T \quad (4)$$

where $\delta \mathbf{x}_S$ is the infinitesimal displacement vector in force sensor space and is measured from strain gauges mounted on the force sensor body; $\mathbf{J}_f(t)$ is the Jacobian mapping and is time-variable; and $\delta \mathbf{X}_T$ is the infinitesimal displacement vector in task space.

Figure 2 shows a typical wrist force sensor mounted at a manipulator end-effector and the associated coordinate frame definitions. Force sensing is based on Hooke's law and is a highly linear process assuming induced strains remain within the elastic range of the material of the force sensor body. Through a quasi-static system stiffness analysis described in [11], the Jacobian mapping can be written as

$$\mathbf{J}_f(t) = \mathbf{C}_S \mathbf{J}_{ST}^{\mathrm{T}} \mathbf{J}_M^{-\mathrm{T}}(\theta)\mathbf{K}_\theta \mathbf{J}_M^{-1}(\theta) \quad (5)$$

where $\mathbf{C}_S$ is the force sensor calibration matrix; $\mathbf{J}_{ST}^{\mathrm{T}}$ is the mapping shown in Figure 2; $\mathbf{J}_M(\theta)$ is the manipulator Jacobian matrix and varies with θ, the vector of joint positions; and $\mathbf{K}_\theta$ is the joint stiffness matrix.

2.4. Comparing Force and Vision Resolvability

In order to perform a comparison of the resolvability of force and vision feedback, the variance of sensor noise must be considered in terms of the resolvability of the sensor. This is given by

$$\sigma_T = \mathbf{J}^{+}(k)\sigma_S \quad (6)$$

where σ_T is the vector of positional variance in task space; $\mathbf{J}^{+}(k)$ is the pseudoinverse of the sensor Jacobian; and σ_S is the vector representing measurement variance in sensor space.

For vision feedback, this variance is dependent on the tracking algorithm used, the size of the feature template, and the quality of the feature being tracked. For the experimental results to be presented, the value is typically around 1.0 pixels. This results in a task positional variance on the order of 0.0003m in a plane parallel to the image plane and 0.003m along the optical axis. The force sensing system used to collect experimental results produces twelve-bit strain gauge readings, which typically have a measured steady-state variance of 2.0 units. The stiffness of the manipulator is derived from the proportional gains on the joints of the manipulator used. These values are on the order of 1000-10000 Nm/rad for the first three Puma joints and 300-500 Nm/rad for the three wrist joints. For a typical configuration far from manipulator singularities, task space positional variances on the order of 10^{-6}m are calculated. Although the sensor is sensitive to displacements in the micron range, the noise introduced by inertial effects during manipulator motion on the strain gauge readings is significantly higher.

3. Port-Based Agents

The desired visual representation of a task is derived from object schemas. This desired visual representation provides reference inputs to a visual servoing feedback loop, shown in Figure 3. Object schemas also determine when vision feedback is unable to resolve relative displacements between objects and force feedback must be used. The vision/force servoing system is a semi-autonomous agent, termed a *port-based agent*, that continuously accepts reference commands from object schemas and attempts to achieve the desired visual representation of the task using visual servoing techniques.

Controlled active vision [13] is used to derive a control strategy for visually servoed port-based agents. A state equation for the visual servoing system can be derived by discretizing (1) and writing this discretized equation as

$$\mathbf{x}(k+1) = \mathbf{x}(k) + T\mathbf{J}_v(k)\mathbf{u}(k) \quad (7)$$

where $\mathbf{x}(k) \in R^{2M}$ and is a vector of feature states, T is the sampling period of the vision system, $\mathbf{u}(k)$ is the commanded manipulator end-effector velocity, and M is the number of features being tracked. Using optimal control techniques, an objective function that places a cost on feature error and control energy of the form

$$F(k+1) = [\mathbf{x}(k+1) - \mathbf{x}_D(k+1)]^{\mathrm{T}}\mathbf{Q}[\mathbf{x}(k+1) - \mathbf{x}_D(k+1)] + \mathbf{u}^{\mathrm{T}}(k)\mathbf{L}\mathbf{u}(k) \quad (8)$$

can be minimized at each time instant to obtain the control law

$$\mathbf{u}(k) = -(T^2\mathbf{J}_v^{\mathrm{T}}(k)\mathbf{Q}\mathbf{J}_v(k) + \mathbf{L})^{-1}T\mathbf{J}_v^{\mathrm{T}}(k)\mathbf{Q}[\mathbf{x}(k) - \mathbf{x}_D(k+1)] \quad (9)$$

The vector $\mathbf{x}_D(k+1)$ represents the desired feature state, i.e. the desired visual representation of the scene, at the next

time instant. This feature vector is derived from the supervisor's actions within the virtual environment. As the supervisor directs controllable object schemas, the projection of the 3D representation of the object schemas through the forward sensor mapping results in $\mathbf{x}_D(k+1)$. $\mathbf{Q}$ and $\mathbf{L}$ are weighting matrices and allow the user to place a varying emphasis on the feature error and the control input. Extensions to this control strategy and guidelines for choosing the matrices $\mathbf{Q}$ and $\mathbf{L}$ can be found in [9]. New sensor configurations can be quickly and easily added to the system by substituting the correct $\mathbf{J}_v$ and by adjusting $\mathbf{Q}$ and $\mathbf{L}$ accordingly.

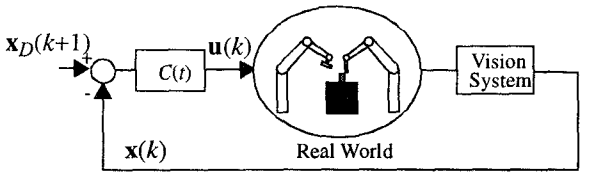

Figure 3. Visual servoing feedback loop.

The measurement of the motion of the features on the image plane must be done continuously and quickly. Our method for measuring this motion is based on optical flow techniques and is a modification of the method proposed in [1]. This technique is known as Sum-of-Squared-Differences (SSD) optical flow, and is based on the assumption that the intensities around a feature point remain constant as that point moves across the image plane. A more complete description of the algorithm and its original implementation can be found in [9].

We have developed a robust force/vision control strategy that compensates for measured forces due to inertial effects induced by end-effector accelerations. The control strategy is discussed in detail in [10] and is written as

$$\dot{\mathbf{x}}_{ref_v} = -(T\mathbf{J}_v^{\mathrm{T}}(k)\mathbf{Q}T\mathbf{J}_v(k) + \mathbf{L})^{-1} T\mathbf{J}_v^{\mathrm{T}}(k)\mathbf{Q}[\mathbf{x}(k) - \mathbf{x}_D(k+1)]$$

$$\dot{\mathbf{x}}_{ref_f} = \mathbf{S}_F\mathbf{G}_F(\mathbf{F}_r(k) - \mathbf{F}_m(k))$$

for each axis, i {

if $(((|\ddot{x}_{m_i}| > \varepsilon_a) \wedge (\dot{x}_{m_i} \mathrm{sgn} F_{m_i} < \varepsilon_v)) \vee$

$(\dot{x}_{ref_{vi}} F_{m_i} > 0.0) \vee (|F_{m_i}| < F_T))$

$\mathbf{S}_v[i,i] = 1.0 \quad \mathbf{S}_F[i,i] = 0.0$

else $\mathbf{S}_v[i,i] = 0.0 \quad \mathbf{S}_F[i,i] = 1.0$

}

$$\mathbf{u}(k) = \mathbf{S}_v\dot{\mathbf{x}}_{ref_v} + \mathbf{S}_r\dot{\mathbf{x}}_r + \mathbf{S}_F\dot{\mathbf{x}}_{ref_f} \qquad (10)$$

where $\mathbf{G}_F$ is a matrix of force control gains; $\mathbf{F}_r$ and $\mathbf{F}_m$ represent reference and measured forces; $\mathbf{S}_v$, $\mathbf{S}_F$, and $\mathbf{S}_r$ are selection matrices; $\dot{x}_{m_i}$ and $\ddot{x}_{m_i}$ represent measured cartesian velocities and accelerations along particular task space directions; ε_a, ε_v, and F_T threshold sensor noise; and $\dot{x}_r$ represents some desired end-effector reference velocity due, for example, to trackball input from a supervisor.

Recently, other researchers have also investigated combining force and vision feedback. In [6] a hybrid adaptive scheme is described in which force and vision servoing occur along orthogonal task directions. Our technique considers force and vision along the same task direction in order to achieve fast, stable contact transitions while generating low impact forces.

4. A Framework of Schemas and Agents

Our system framework of schemas and port-based agents for telerobotic manipulation draws many of its concepts from past work in the development of expectation-based and verification approaches for guiding mobile robots. An expectation-based approach to scene understanding was first explicitly proposed by Dickmanns [2]. His work was originally concerned with guiding autonomous mobile systems in rapidly changing environments, particularly autonomous vehicles and aircraft. Roth and Jain propose a "verification-based" approach to navigation in the world [15]. A key point of both the expectation and verification-based approaches is that strong internal models of the recent world state are maintained. This significantly reduces the number of hypotheses that must be considered when determining the current state of the world based on sensory data. Neisser's view of the human "perceptual cycle" is similar in many ways to a verification or expectation-based approach [7][12]. Figure 4 shows a modified representation of Neisser's "perceptual cycle." This figure illustrates our view of the relationship between object schemas of the world, the real world, and port-based agents. The counter-clockwise flow of information represents the cyclical nature of the system; sensory data updates schemas, which in turn provide inputs to port based agents, which provide sensory data obtained from the real world to the object schemas existing within the environment model. This cycle illustrates the interaction between perception of the world, actions taken within this world, and plans made about the world.

It is within the environment modeler that the object schemas exist. A supervisor guides CAD model representations of object schemas with a mouse in order to describe the task. Figure 5 shows a graphical representation of the internal schema representation and the corresponding live video imagery. This is similar to the remote teleoperative techniques described in [4] and [5].

5. Hardware Implementation

The schema-agent framework has been implemented on a robotic assembly system consisting of three Puma 560's called the Troikabot. An Adept robot is also used for providing accurate target motion for experimental purposes. The workcell is shown in Figure 6. A Datacube Maxtower Vision System calculates the optical flow of the features

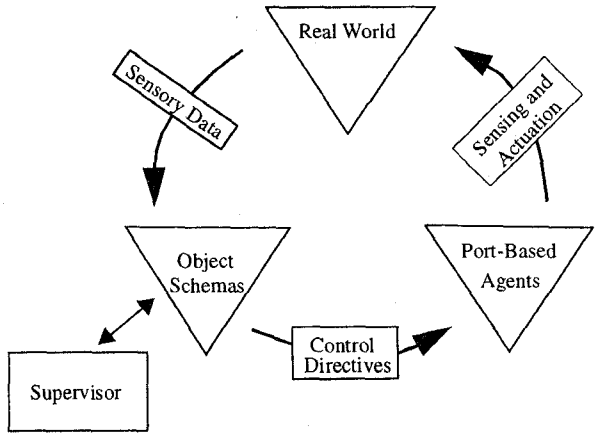

Figure 4. A modified "perceptual cycle."

using an SSD optical flow algorithm. An image can be grabbed and displacements for up to five 16x16 features in the scene can be determined at 30Hz. Stereo system implementations result in half the sampling frequency because only a single digitizer exists on the Datacube. For the stereo system with which experimental results were obtained, visual servoing is performed at 15Hz while tracking ten features. The vision system VME communicates with the robot controller VME using BIT3 VME-to-VME adapters. The dynamic environment model runs on a Silicon Graphics Crimson. The modeler communicates with the VME-based robot controllers via a Bit3 VME-VME hardware adapter. This allows direct shared memory communication between the motion planner and the manipulators running under the Chimera real-time operating system [17].

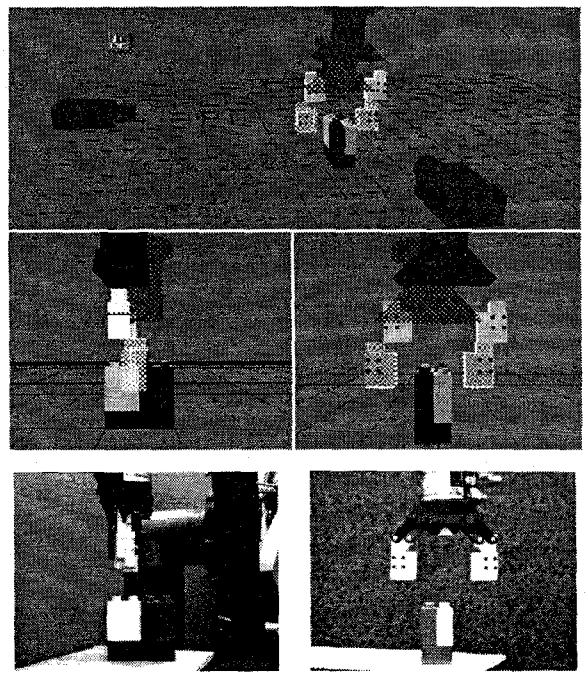

Figure 5. Teleoperator interface to object schemas and corresponding "live" camera views.

6. Experimental Results

A set of experimental results was collected in order to illustrate the advantages of our proposed feedback assimilation strategy. A supervisor was given the task of mating two rigid objects quickly, stably, and with low impact forces. The supervisor was allowed to view the world only from within the virtual environment. Motion of an object held by a manipulator was directed by dragging the graphical representation of the object with a mouse. The port-based agent performing the motion used the force/vision control algorithm described by (10). For comparison purposes, two other common impact strategies that use only force feedback, a guarded move and damping force control, were also tested. Figure 7 shows an experimental comparison of the three strategies. The solid line shows the performance of the force/vision servoing algorithm. The supervisor directs one object toward the other as quickly as possible. Initially, the two objects are 5.9cm from one another. A force of -2N between the two objects is maintained after contact. This strategy achieves contact after 1.43s, and achieves a stable -2N contact force after approximately 4.5s. With simple damping force control alone (dash-dotted line), the distance of 5.9cm is covered in 3.1s. As soon as the objects impact with one another the manipulator becomes unstable. The only way to achieve stable contact using damping control alone given the force control implementation used, is to reduce the force gains to extremely low values, resulting in unacceptably slow motion. Figure 7 also shows a force plot of a guarded move in which the force sensor is monitored at 100Hz. The guarded move strategy allowed only moderate speeds (0.02m/s) and still resulted in high impact forces. At higher speeds, extremely high impact forces would result that could have easily damaged the manipulator. High con-

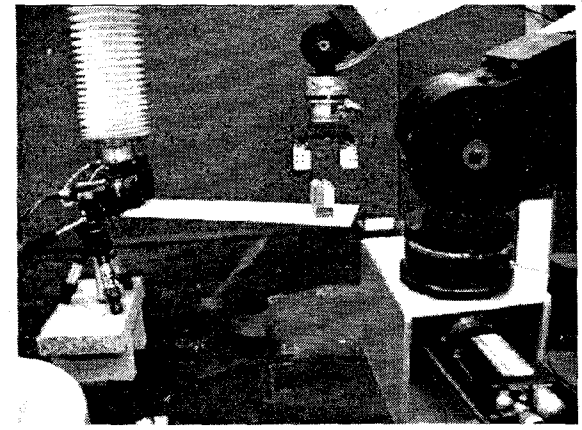

Figure 6. Workcell view

tact forces are created because of the finite amount of time required to stop the end-effector after contact is sensed illustrating the main limitation of a guarded move strategy. For these experimental results force gains of 0.001m/(Ns) were used, the diagonal elements of **Q** were chosen to be $2.0\text{x}10^{-6}$, and the diagonal elements of **L** were chosen to be 10.0. Thresholds were chosen to be ε_a=0.01m/s^2, ε_v=0.001m/s, and F_T=1.5N.

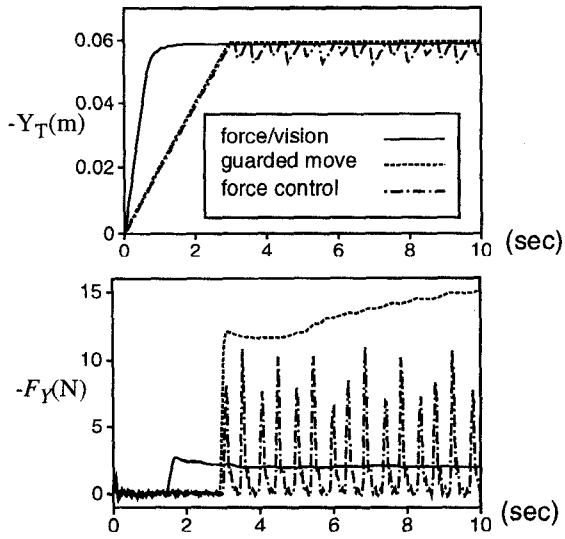

Figure 7. Vertical position and measured force for force/vision control, a guarded move, and damping force control.

7. Conclusion

Teleoperation often requires the use of disparate sensing modalities. In this paper, we have described a framework of objects schemas and port-based agents. Object schemas use the concept of vision and force sensor resolvability as a means of comparing the ability of the two sensing modes to provide useful information during telerobotic manipulation within a virtual environment. By monitoring the resolvability of the two sensing modes with respect to the environment model, the information provided by the disparate sensors can be seamlessly assimilated during teleoperated manipulation. A port-based agent that employs a nonlinear force/vision servoing algorithm to switch between sensing modes demonstrates the advantages of the assimilation technique. Contact transitions between a stiff manipulator and rigid environment, a system configuration that easily becomes unstable when force control alone is used, are robustly achieved. Experimental results show that the nonlinear controller is able to simultaneously satisfy the conflicting task requirements of fast approach velocities, maintaining stability, minimizing impact forces, and suppressing bounce between contact surfaces. The proper assimilation of force and vision feedback is the key to the success of this strategy.

8. Acknowledgments

This research was supported in part by the U.S. Army Research Office through Grant Number DAAL03-91-G-0272, by Sandia National Laboratories through Contract Number AC-3752D, and by the National Science Foundation through Grant Numbers IRI-9612329 and CDA-9616757.

9. References

[1] P. Anandan, "A Computational Framework and an Algorithm for the Measurement of Visual Motion," *Int. J. of Computer Vision*, 2(3), pp. 283-310, January, 1989.

[2] E.D. Dickmanns, "Expectation-based Dynamic Scene Understanding," *Active Vision*, eds. A. Blake and A. Yuille, pp. 303-335, The MIT Press, Cambridge, 1992.

[3] H.F. Durrant-Whyte, *Integration, Coordination and Control of Multi-Sensor Robot Systems*, Kluwer Academic Publishers, 1988.

[4] C. Fagerer, D. Dickmanns, and E.D. Dickmanns, "Visual Grasping with Long Delay Time of a Free Floating Object in Orbit," *Autonomous Robots*, 1(1), pp. 53-68, 1994.

[5] G. Hirzinger, "ROTEX-the first space robot technology experiment," *Experimental Robotics III: The Third Int. Symp., Kyoto, Japan, Oct. 28-30, 1993*, eds. T. Yoshikawa and F. Miyazaki, Springer-Verlag, pp. 579-598, 1994.

[6] K. Hosoda, K. Igarashi, and M. Asada, "Adaptive Hybrid Visual Servoing/Force Control in Unknown Environment," *Proc. 1996 IEEE/RSJ Int. Conf. on Intelligent Robots and Sys. (IROS96)*, Osaka, 1996.

[7] R. Jain, "Environment Models and Information Assimilation," Technical Report RJ 6866(65692), IBM-Yorktown Heights, 1989.

[8] V.C. Klema, and A.J. Laub, "The Singular Value Decomposition: Its Computation and Some Applications," *IEEE Tran. on Automatic Control*, 25(2), pp. 164-176, 1980.

[9] B. Nelson, N.P. Papanikolopoulos, and P.K. Khosla, "Visual Servoing for Robotic Assembly," *Visual Servoing—Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, ed. K. Hashimoto, World Scientific, New Jersey, pp. 139-164, 1993.

[10] B.J. Nelson, J.D. Morrow, and P.K. Khosla, "Improved Force Control Through Visual Servoing," *Proc. 1995 American Control Conference (ACC95)*, pp. 380-386, Seattle, June, 1995.

[11] B.J. Nelson and P.K. Khosla, "Force and Vision Resolvability for Assimilating Disparate Sensory Feedback," *IEEE Trans. on Robotics and Automation*, 12(5), pp. 714-731, October, 1996.

[12] U. Neisser, *Cognition and Reality*, W.H. Freeman and Co., New York, 1976.

[13] N.P. Papanikolopoulos, P.K. Khosla, and T. Kanade, "Adaptive Robotic Visual Tracking," *Proc. of the American Control Conference*. Evanston, IL.:American Autom. Control Council, pp. 962-967, 1991.

[14] J.M. Richardson and K.A. Marsh, "Fusion of Multisensor Data," *Int. J. of Robotics Research*, 7(6), pp. 78-96, 1988.

[15] Y. Roth and R. Jain, "Integrated Architectures for Autonomous Systems," *Proc. of the SPIE - The Int. Society for Optical Engineering*, vol. 1571, pp. 628-639, 1991.

[16] R.C. Smith and P. Cheeseman, "On the Representation and Estimation of Spatial Uncertainty. *Int. J. of Robotics Research*, 5(4), pp. 56-68, 1986.

[17] D.B. Stewart, D.E. Schmitz, and P.K. Khosla, "The Chimera II Real-Time Operating System for Advanced Sensor-Based Control Systems," *IEEE Trans. Sys., Man and Cyb.*, 22, pp. 1282-1295, 1992.