Technical Report

Force and Vision Resolvability for Assimilating Disparate Sensory Feedback

Bradley J. Nelson Pradeep K. Khosla

CMU-RI-TR-95-11

The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3890

February 1995

© 1995 Carnegie Mellon University

This research was supported in part by the U.S. Army Research Office through Grant Number DAAL03-91-G-0272 and by Sandia National Laboratories through Contract Number AC-3752D. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies.

Abstract

Force and vision sensors provide complementary information, yet they are fundamentally different sensing modalities. This implies that traditional sensor integration techniques that require common data representations are not appropriate for combining the feedback from these two disparate sensor. In this paper, we introduce 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 robotic manipulation tasks. By monitoring the resolvability of the two sensing modes with respect to the task, the information provided by the disparate sensors can be seamlessly assimilated during task execution. A nonlinear force/vision servoing algorithm that uses force and vision resolvability to switch between sensing modes is proposed. The advantages of the assimilation technique is demonstrated during contact transitions between a stiff manipulator and rigid environment, a system configuration that easily becomes unstable when force control alone is used. Experimental results show that robust contact transitions are made by the proposed nonlinear controller while simultaneously satisfying the conflicting task requirements of fast approach velocities, maintaining stability, minimizing impact forces, and suppressing bounce between contact surfaces.

Table of Contents

1. Introduction 1
2. Previous Work 3
    2.1 Force Servoing 3
    2.2 Visual Servoing 3
    2.3 Sensor Resolution 5
    2.4 Combining Force and Vision Feedback 6
3. Resolvability as a Measure of Sensor Disparateness 7
    3.1 Vision Resolvability 7
        3.1.1 Monocular Systems 7
        3.1.2 Binocular Systems with Parallel Optical Axes 10
        3.1.3 Binocular Systems with Perpendicular Optical Axes 12
        3.1.4 Vision Resolvability Ellipsoids 12
    3.2 Force Resolvability 15
    3.3 Comparing Vision and Force Resolvability 19
4. Visual Servoing Formulation 21
    4.1 Controller 21
    4.2 Feature Tracking 22
5. Vision/Force Servoing 24
6. Experimental Results 28
    6.1 Hardware Setup 28
    6.2 Results 28
7. Conclusion 36
8. References 37

List of Figures

Figure 1: Hybrid force/position control loop. 4
Figure 2: A visual servoing control loop. 4
Figure 3: Task frame and camera frame definitions. 8
Figure 4: Task frame-camera frame definitions for a binocular system with parallel optical axes. 11
Figure 5: Task frame-camera frame definitions for a binocular system with perpendicular optical axes. 12
Figure 6: Resolvability Ellipsoids: monocular system, f=24mm, depth=1.0m, 2 features located in the task frame at (0.1m,0.1m,0) and (-0.1m,0.1m,0). 13
Figure 7: Resolvability Ellipsoids: monocular system, f=12mm, depth=0.5m, 2 features located in the task frame at (0.1m,0.1m,0) and (-0.1m,0.1m,0). 13
Figure 8: Resolvability of depth versus depth of object and focal length for two features located in the task frame at (0.05m,0,0) and (-0.05m,0,0). 14
Figure 9: Resolvability in orientation about Z versus center of object projection onto the image plane. 14
Figure 10: Resolvability Ellipsoids: stereo pair-parallel optical axes, f=12mm, b=20cm, depth=1.0m, 1 feature located in the task frame at (0,0.2m,0). 14
Figure 11: Resolvability in depth versus baseline length and depth of object for a stereo pair, parallel optical axes, f=12mm, and a single feature located at the origin of the task frame. . . 15
Figure 12: Resolvability Ellipsoids: stereo pair-perpendicular optical axes, f=12mm, depth=1.0m, 2 features located in the task frame at (-0.1m,0.1m,0), and (0.1m,-0.1m,-0.1m). 15
Figure 13: Coordinate frame definitions for a manipulation task that employs force sensing . 16
Figure 14: An augmented form of Kim's Premultiplier Diagram (Kim et al 1992) for illustrating transformations between force sensor strain gauge measurements and end-effector cartesian displacements under quasi-static conditions. 18
Figure 15: A pyramidal search scheme is used in the SSD optical flow algorithm in order to increase the overall sampling rate of the system. 23
Figure 16: Force and vision in the feedback loop. 25
Figure 17: Inertial forces measured by the force sensor and the corresponding measured cartesian accelerations which induced these forces. 27
Figure 18: Laboratory setup used for performing vision/force servoing experiments. 27
Figure 19: The Troikabot system architecture. 29
Figure 20: The camera view for visual servoing and coordinate axes on the image plane and in

the task frame. 29
Figure 21: Vertical error between desired and measured end-effector position on the image plane for three different trials, each with a different error in the estimated location of the surface. 30
Figure 22: Vertical position of end-effector in cartesian space and force measured along the vertical direction versus time for all three trials. 31
Figure 23: Commanded end-effector cartesian velocity along -Y for the three trials with varying error magnitudes in the estimated surface location.. 32
Figure 24: Vertical position of end-effector in cartesian space, force measured along the vertical direction, and pixel error versus time for traded control. 33
Figure 25: Vertical position of end-effector in cartesian space and force measured along the vertical direction versus time for simple damping force control upon impact with a surface. . 34
Figure 26: Vertical position of end-effector in cartesian space and force measured along the vertical direction versus time for a guarded move impact strategy that switches to position control. 34
Figure 27: Combined plots of vertical position and measured force during impact for force/vision control, damping force control, and a guarded move. 35

1. Introduction

Within the domain of robotic manipulation, roboticists have traditionally considered force feedback to be the most relevant sensing modality. This is because of the need for highly accurate information on the relative positions of objects and on the nature of contact forces between objects being manipulated. More recently, many researchers have realized the benefits of using visual servoing techniques to reduce alignment uncertainties between objects using imprecisely calibrated camera-lens-manipulator systems. These two sensing modalities, force and vision, are complementary in the sense that they are useful during different stages of task execution; vision brings parts into alignment; force ensures reasonable contact forces are maintained as parts mating occurs.

Force and vision are complementary sensing modalities because of their disparate nature. The output of a typical manipulator wrist force sensor yields measurements of force and torque along and about the three cartesian axes. Properly tracked features in a visual sensor's sensing space yield measurements of the relative positions and orientations of objects in the world. The two sensing systems produce fundamentally different quantities, force and position.[1] This presents an inherent problem when attempting to integrate information from the two sensors. Sensor integration techniques require a common representation among the various sensory data being integrated. Force and vision sensors do not provide this common data representation. Furthermore, the two sensing modes are useful during different stages of the task being executed because of their disparate natures. These two facts indicate that traditional sensory integration techniques are not appropriate.

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, for example (Smith and Cheeseman 1986; Durrant-Whyte 1988; Richardson and Marsh 1988; Hager 1990). This has obvious drawbacks for integrating force and vision feedback, 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. Vision and force sensing modes are appropriate during different stages of the task, making a temporal comparison of the two data sets meaningless during most of the task.

Instead, we propose a task oriented technique for assimilating information from force and vision sensors. We believe in the importance of the task model for combining information from disparate sensors, much as (Jain 1989) argues for the importance of environment models for the assimilation of information from disparate sensors in a mobile robot domain. It makes little sense to combine the measurements of force and position using, for example, a Kalman filter because of the disparate

[1] Throughout this paper the term "force" refers to "force and torque" and "position" refers to "position and orientation," unless otherwise noted.

nature of the feedback. A model of the task is required which has the capability of dynamically assimilating information from the two disparate sensing modes. As the task occurs, the task model determines when vision is appropriate and when force is appropriate by considering the nearness of contact surfaces and the resolution with which each sensor can sense object locations.

Throughout a manipulation task, the data from both sensors must be compared in order to ascertain which sensing mode is more relevant to the task at the given time instant. Our previous work in *resolvability* (Nelson and Khosla 1994a; Nelson and Khosla 1994b) showed how various visual sensing configurations can be compared in terms of the resolution with which they can be used to visually servo an object held by a manipulator. In this paper, we extend this concept to force sensors in order to determine the resolution with which a force sensor can detect infinitesimal task displacements in the environment. *Force resolvability* is dependent not only on the force sensor, but also on the stiffness of the entire system with respect to task displacements. This extension of resolvability provides a common measure for both sensors for evaluating when visual servoing or force servoing strategies are appropriate.

During transitions between force and vision sensing, a nonlinear force/vision control law compensates for the uncertain world until it becomes clear when a new sensing mode has been achieved, or whether the system should return to the prior sensing mode. In order to illustrate the advantages of assimilating disparate sensor feedback using our proposed method, we experimentally demonstrate the performance of the technique during contact transitions. Many researchers have studied the impact problem, and various impact strategies have been proposed. However, the fundamental problem of using force feedback alone to minimize impact forces while quickly achieving contact stably within imprecisely calibrated environments still exists. By combining vision feedback with force feedback using the concept of resolvability and our proposed nonlinear control strategy, we demonstrate how fast stable contact transitions with a stiff manipulator in a rigid environment can be achieved.

In this paper, we demonstrate the use of force and vision resolvability for assimilating high bandwidth visual feedback (30Hz) and high bandwidth force feedback (100Hz) within the same manipulator feedback loop. After reviewing related work, we discuss the concept of vision and force resolvability and how they can be used in the sensor assimilation process. Next, a visual servoing control law is derived, followed by a description of the vision/force servoing control strategy. An important contribution of the work to be presented in this paper is that we show how vision can be used to greatly simplify the stability problem by allowing the effective use of low gain force control with stiff manipulators (a Puma 560). Since the stability of low gain force control is much easier to maintain, the use of force feedback during manipulator fine motion is more easily realized because simple force control strategies can be used without the need for high order models of the arm, sensor, and environment for choosing stable controller gains. The proper combination of force and vision feedback is the key to the success of this strategy.

2. Previous Work

2.1 Force Servoing

Robotic force control has been studied since the 1950's. A survey on the history of force control can be found in (Whitney 1985). Active impedance control has been suggested as the most general form of force control (Hogan 1985), however, difficulties in programming impedance controlled manipulators have resulted in very limited use of this strategy. Hybrid control (Raibert and Craig 1981) separates position control and force control into two separate control loops that operate in orthogonal directions, as shown in Figure 1, and was extended by (Yoshikawa 1987) to include manipulator dynamics.

One of the most important issues in force control is maintaining manipulator stability (Whitney 1985). Force controllers must be properly formulated and tuned in order to maintain stability, and this can be difficult, particularly during initial contact between stiff surfaces. During impact, another important issue is the generation of large impact forces. (Volpe and Khosla 1991) demonstrated an effective impact strategy based on a proportional gain explicit force controller with a feedforward signal and negative gains. The gains for the controller were chosen using a fourth order model of the arm, sensor, and environment in which a frictionless arm was assumed (experiments were conducted with the CMU Direct-Drive Arm II). Although extremely high impact velocities were achieved (75cm/s), large impact forces were also generated (90N). This illustrates a typical problem exhibited by all force control strategies during impact with rigid objects, for example (Khatib and Burdick 1986; An and Hollerbach 1987; Eppinger and Seering 1987; Hogan 1987; Kazerooni 1987; Qian and De Schutter 1992; Hyde and Cutkosky 1993; Xu, Hollerbach, and Ma 1994): high impact velocities, manipulator stability, low impact forces, and quickly achieving the desired force are all contradictory system requirements.

2.2 Visual Servoing

Visual servoing has a less extensive history than that of force control, mainly due to the lack of computational resources available for processing the large amounts of data contained in an image. Although previous researchers had considered fast visual feedback for guiding manipulator motion, for example (Shirai and Inoue 1973), the visual servoing field was first well defined by (Weiss 1984). Since the work by Weiss, two types of visual servoing configurations have emerged, eye-in-hand configurations and static camera configurations. Eye-in-hand visual servoing tracks objects of interest with a camera mounted on a manipulator's end-effector (Weiss, Sanderson and Neumann 1987; Allen 1989; Corke and Paul 1989; Feddema and Lee 1990; Ghosh 1992; Papanikolopoulos, Khosla and Kanade 1991; Espiau, Chaumette and Rives 1992; Hashimoto and Kimura 1993; Wilson 1993). Static camera visual servoing guides manipulator motion based on feedback from a camera observing the end-effector (Koivo and Houshangi 1991; Nelson, Papanikolopoulos

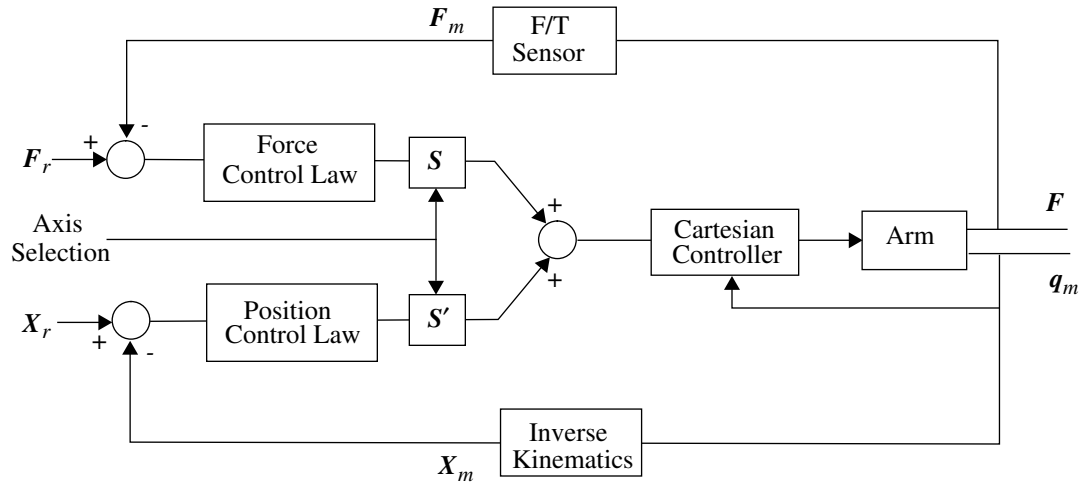

Figure 1: Hybrid force/position control loop, where F_r is the reference force vector, F_m is the measured force vector, X_r is the reference position, X_m is the measured position, S and S' are the orthogonal selection matrices, F is the applied force, and q_m is a vector of measured joint positions.

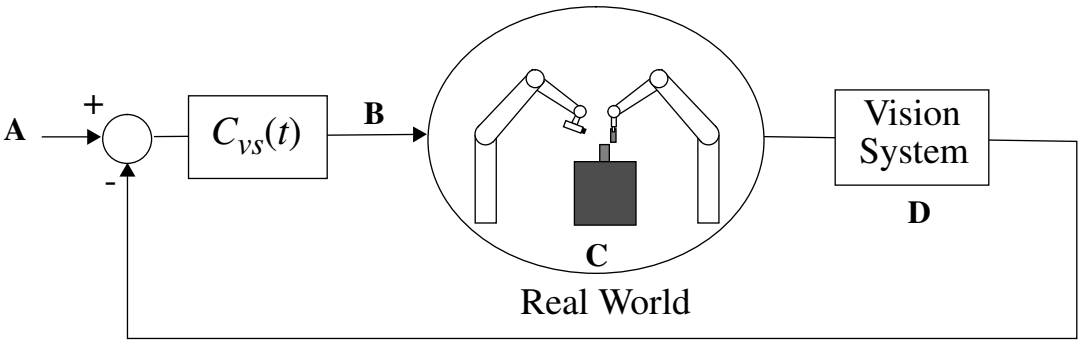

Figure 2: A visual servoing control loop. Differences in proposed control schemes include: A-reference inputs given in cartesian or sensor coordinates, the dimensionality of the control space; B-manipulator commands given by position or velocity setpoints or torques; C-eye-in-hand or static camera configurations; and D-feature tracking algorithms used.

and Khosla 1993; Castaño and Hutchinson 1994; Hager 1994). Most of this past work has been with monocular systems, though recently stereo systems have been used for visual servoing (Maru et al. 1993; Hager 1994; Hosoda and Asada 1994).

A typical visual servoing feedback loop is shown in Figure 2. Differences between the various approaches to visual servoing include the space in which reference inputs are provided, the dimensionality of the control space, the structure of the controller, the physical configuration of the system, the derivation of the control law, and the feature tracking algorithms used. An excellent survey of recent work in visual servoing can be found in (Corke 1993).

2.3 Sensor Resolution

The concept of sensor resolution plays an important role in the assimilation of force and vision feedback. In order to effectively use visual feedback to perform robotic tasks, many researchers have recognized that the placement of the sensor relative to the task is an important consideration, and sensor resolution has been considered in the past as a criterion for sensor planning (Cowan and Kovesi 1988; Tarabanis et al 1990; Yi et al 1990). These efforts concern static camera systems in which a required spatial resolution is known and a single camera placement is desired. In (Das and Ahuja 1993), a study of stereo, vergence, and focus cues for determining range is described in which the performance of each cue for determining range accuracy is characterized. This characterization can be used to control camera parameters in order to improve the accuracy of range estimates. Our resolvability approach can be used for determining the ability of a visually servoed manipulator to accurately resolve positions and orientations of objects along all six degrees of freedom. Resolvability provides a technique for estimating the relative ability of various visual sensor systems, including single camera systems, stereo pairs, multi-baseline stereo systems, and 3D rangefinders, to accurately control visually manipulated objects and to provide spatially accurate data on objects of interest. Camera-lens intrinsic and extrinsic parameters can be actively controlled using a resolvability measure in conjunction with other sensor placement criteria so that the accuracy of visual control can be improved (Nelson and Khosla 1994a). The concept can also be used for static sensor placement for either object recognition or visual servoing.

(Sharma and Hutchinson 1994) introduced a measure similar to resolvability that they call *observability* (though unrelated to observability in the controls sense). Their proposed algorithm attempts to maximize $\sqrt{det\,(\mathbf{J}\mathbf{J}^{\mathrm{T}})}$ throughout a camera trajectory, where $\mathbf{J}$ is the image Jacobian. This measure is non-zero only when the dimension of the feature space is exactly equal to the dimension of the task space and the features are configured in a non-singular fashion. Our emphasis on resolvability has been in its directional nature, as determined from the singular values of $\mathbf{J}$ and the eigenvectors of $\mathbf{J}^{\mathrm{T}}\mathbf{J}$, and in using this decomposition to actively guide camera-lens motion during task execution (Nelson and Khosla 1994a). In this paper, we extend resolvability to include force sensors. This measure is then used to assimilate force and vision information within high bandwidth manipulator feedback loops.

For force sensor design, strain gauge sensitivity, force sensitivity, and minimum sensor stiffness are three critical design parameters. In (Nakamura et al 1988) and (Uchiyama et al 1991), force sensor design techniques are based on measures derived in part from a singular value decomposition of the force sensor calibration matrix. A measure of force sensor resolution to be presented in Section 3.2 uses this past work as a foundation and extends the analysis to include system stiffness as well.

2.4 Combining Force and Vision Feedback

Many researchers have considered combining force and vision feedback, though very few have done this within high bandwidth feedback control loops. One of the first papers to mention the benefits of combining high bandwidth visual and force feedback is by (Shirai and Inoue 1973). They implemented a 0.1Hz visual servoing scheme and referred to the use of force servoing, but a lack of computational resources hampered their effort, and many of the issues of combining the two sensing modalities went unnoticed. In (Ishikawa, Kosuge and Furuta 1991), visual servoing of 2 Hz was used to align a wrench with a bolt before a compliant wrenching operation is performed. Again, vision and force were not explicitly combined, and the issues concerning their integration remained unaddressed.

(Durrant-Whyte 1987) proposed a Bayesian framework for combining visual observations and tactile data for grasping. The non-dynamic nature of the task made the use of the tactile feedback of questionable value for the experiments described. (Allen 1988) and (Stansfield 1988) both proposed rule-based approaches in order to combine vision and touch feedback for object recognition. An important observation made in (Allen 1988) is that touch feedback is successful for object recognition because vision provides cues to the tactile sensor. This is also important for object manipulation, and is a primary reason why fast, stable contact transitions can be more easily realized with manipulators servoed under both vision and force control, rather than force control alone. The assimilation technique proposed in this paper employs a nonlinear control strategy that is a combination of quantitative and rule-based approaches of combining force and vision sensing. The general idea behind Durrant-Whyte's (1988) quantitative multi-sensor integration framework is used to determine the appropriate sensor at any given instant. Rule-based methods are used to perform mode switching and to eliminate false force sensor readings.

3. Resolvability as a Measure of Sensor Disparateness

All sensor fusion techniques that use multiple sensors along the same task dimensions require that the system must compare the characteristics of the feedback from each individual sensor at some point during the task. For sensors that have similar data representations, for example multiple cameras that provide positional information, this is straightforward. For sensors that provide fundamentally different measurements, for example force and vision sensors, this presents a problem. Our past work in resolvability demonstrated how sensor resolution can be used to compare the effects of camera-lens-task configurations on the ability to accurately resolve, and therefore visually servo, object positions and orientations (Nelson and Khosla 1994b). In this paper, we extend our past work in resolvability for camera-lens systems, which we now refer to as *vision resolvability,* to include the concept of *force resolvability*. This provides a measure of the ability of both force and vision sensors to resolve positions and orientations in task space, thus providing a method 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, and $\delta \mathbf{X}_T$ is an infinitesimal displacement vector in task space.

By performing a singular value decomposition (Klema and Laub 1980) on the task space to sensor space Jacobian, and analyzing the singular values of $\mathbf{J}$ and the eigenvectors of $\mathbf{J^T J}$ 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.

3.1 Vision Resolvability

3.1.1 Monocular Systems

3.1.1.1 Camera Model

The mapping from task space to sensor space for any system using a camera as the visual sensor requires a camera-lens model in order to represent the projection of task objects onto the CCD image plan. For visual servoing, a simple pinhole camera model has proven adequate for visual tracking using our experimental setup. If we place the camera coordinate frame {C} at the focal point

of the lens as shown in Figure 3, a feature on an object at ${}^{C}\mathbf{P}$ with coordinates (X_C, Y_C, Z_C) in the camera frame projects onto the camera's image plane at

$$x_i = \frac{fX_C}{s_x Z_C} + x_p \tag{2}$$

$$y_i = \frac{fY_C}{s_y Z_C} + y_p \tag{3}$$

where (x_i,y_i) are the image coordinates of the feature, f is the focal length of the lens, s_x and s_y are the horizontal and vertical dimensions of the pixels on the CCD array, and (x_p,y_p) is the piercing point of the optical axis on the CCD. This model assumes that $|Z_C| \gg |f|$.

The mapping from camera frame feature velocity to image plane optical flow, or sensor space velocity, can be obtained simply by differentiating (2) and (3). This yields the following equations

$$\dot{x}_S = \frac{f\dot{X}_C}{s_x Z_C} - \frac{fX_C\dot{Z}_C}{s_x Z_C^2} = \frac{f\dot{X}_C}{s_x Z_C} - x_S\frac{\dot{Z}_C}{Z_C} \tag{4}$$

$$\dot{y}_S = \frac{f\dot{Y}_C}{s_y Z_C} - \frac{fY_C\dot{Z}_C}{s_y Z_C^2} = \frac{f\dot{Y}_C}{s_y Z_C} - y_S\frac{\dot{Z}_C}{Z_C} \tag{5}$$

where $x_S = x_i - x_p$ and $y_S = y_i - y_p$. This defines the mapping from the camera frame onto the image plane. The next step is to transform task space velocities into the camera frame, and then project these camera frame velocities onto the sensor space to obtain the mapping from task space velocity to sensor space velocity.

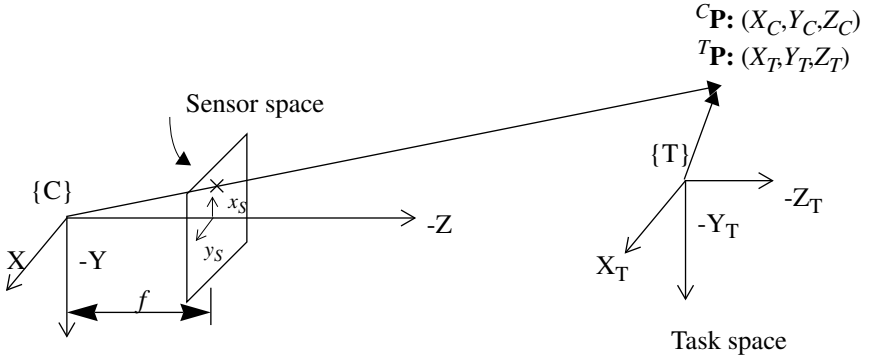

Figure 3: Task frame and camera frame definitions.

3.1.1.2 Objects Defined in a Task Frame

For visually servoing a manipulator holding an object, the objective is to move the image coordinates of ${}^{C}\mathbf{P}$ to some location on the image plane by controlling the motion of ${}^{C}\mathbf{P}$. Typically, ${}^{C}\mathbf{P}$ is some feature on an object being held by a manipulator. Thus, the motion of ${}^{C}\mathbf{P}$ is induced relative to the tool frame of the manipulator being observed. Figure 3 shows the coordinate systems used to define the mapping from task space to sensor space for ${}^{T}\mathbf{P}$ with coordinates in the task frame of

(X_T,Y_T,Z_T). For now, we assume that the rotation of the task frame {T} with respect to {C} ${}^{C}_{T}\mathbf{R}$ is known. The velocity of ${}^{C}\mathbf{P}$ can be written as

$$ {}^{C}\dot{\mathbf{P}} = {}^{C}_{T}\mathbf{R}\,({}^{T}\mathbf{V} + {}^{T}\dot{\mathbf{P}} + {}^{T}\Omega \times {}^{T}\mathbf{P}) \tag{6} $$

where ${}^{T}\mathbf{V} = \begin{bmatrix} \dot{X}_T & \dot{Y}_T & \dot{Z}_T \end{bmatrix}^{\mathrm{T}}$ and ${}^{T}\Omega = \begin{bmatrix} \omega_{X_T} & \omega_{Y_T} & \omega_{Z_T} \end{bmatrix}^{\mathrm{T}}$ are the translational and rotational velocities of the task frame with respect to itself, respectively. These are manipulator end-effector velocities that can be commanded. Since the object being servoed or observed is rigidly attached to the task frame, ${}^{T}\dot{\mathbf{P}} = 0$, and (6) becomes

$$ {}^{C}\dot{\mathbf{P}} = {}^{C}_{T}\mathbf{R}\,({}^{T}\mathbf{V} + {}^{T}\Omega \times {}^{T}\mathbf{P}) \tag{7} $$

Furthermore, if we assume that {C} and {T} are aligned, as shown in Figure 3, then ${}^{C}_{T}\mathbf{R} = \mathbf{I}$, the identity matrix, and the elements of ${}^{C}\dot{\mathbf{P}}$ can be written as

$$ \begin{aligned} \frac{dX_C}{dt} &= \dot{X}_T + Z_T\omega_{Y_T} - Y_T\omega_{Z_T} \\ \frac{dY_C}{dt} &= \dot{Y}_T - Z_T\omega_{X_T} + X_T\omega_{Z_T} \\ \frac{dZ_C}{dt} &= \dot{Z}_T + Y_T\omega_{X_T} - X_T\omega_{Y_T} \end{aligned} \tag{8} $$

The assumption that {C} and {T} are aligned is only used in formulating the Jacobian from task space to sensor space. If the transformation from task space to sensor space is initially known, and the commanded task frame velocity is known, then the coordinates (X_T,Y_T,Z_T) can be appropriately updated while visual servoing. It will also be necessary to account for task frame rotations when determining the velocity to command the task frame based on ${}^{T}\mathbf{V} = \begin{bmatrix} \dot{X}_T & \dot{Y}_T & \dot{Z}_T \end{bmatrix}^{\mathrm{T}}$ and ${}^{T}\Omega = \begin{bmatrix} \omega_{X_T} & \omega_{Y_T} & \omega_{Z_T} \end{bmatrix}^{\mathrm{T}}$. It would have been possible to include the terms of ${}^{C}_{T}\mathbf{R}$ in (8), however, the assumption made simplifies the derivation and does not affect the end result.

By combining (8) with (4) and (5), the entire Jacobian transformation for a single feature from task space to sensor space can now be written in the form

$$ \begin{bmatrix} \dot{x}_S \\ \dot{y}_S \end{bmatrix} = \begin{bmatrix} \frac{f}{s_x Z_C} & 0 & -\frac{x_S}{Z_C} & -\frac{Y_T x_S}{Z_C} & \left[\frac{fZ_T}{s_x Z_C} + \frac{X_T x_S}{Z_C}\right] & -\frac{fY_T}{s_x Z_C} \\ 0 & \frac{f}{s_y Z_C} & -\frac{y_S}{Z_C} & -\left[\frac{fZ_T}{s_y Z_C} + \frac{Y_T y_S}{Z_C}\right] & \frac{X_T y_S}{Z_C} & \frac{fX_T}{s_y Z_C} \end{bmatrix} \begin{bmatrix} \dot{X}_T \\ \dot{Y}_T \\ \dot{Z}_T \\ \omega_{X_T} \\ \omega_{Y_T} \\ \omega_{Z_T} \end{bmatrix} \tag{9} $$

For the above form of the Jacobian, the parameters of the Jacobian are given by $\phi = (f, s_x, s_y, x_S, y_S, Z_C, X_T, Y_T, Z_T)$. Alternatively, the sensor coordinates may be omitted and replaced with camera frame coordinates to arrive at a Jacobian of the form

$$\begin{bmatrix} \dot{x}_S \\ \dot{y}_S \end{bmatrix} = \begin{bmatrix} \frac{f}{s_x Z_C} & 0 & -\frac{fX_C}{s_x Z_C^2} & -\frac{fX_C Y_T}{s_x Z^2{}_C} & \left[\frac{fZ_T}{s_x Z_C} + \frac{fX_C X_T}{s_x Z^2{}_C}\right] & -\frac{fY_T}{s_x Z_C} \\ 0 & \frac{f}{s_y Z_C} & -\frac{fY_C}{s_y Z_C^2} & -\left[\frac{fZ_T}{s_y Z_C} + \frac{fY_C Y_T}{s_y Z^2{}_C}\right] & \frac{fY_C X_T}{s_y Z^2{}_C} & \frac{fX_T}{s_y Z_C} \end{bmatrix} \begin{bmatrix} \dot{X}_T \\ \dot{Y}_T \\ \dot{Z}_T \\ \omega_{X_T} \\ \omega_{Y_T} \\ \omega_{Z_T} \end{bmatrix} \tag{10}$$

where the parameters are now $\phi = (f, s_x, s_y, X_C, Y_C, Z_C, X_T, Y_T, Z_T)$. Either form may be desirable depending on the design parameters desired for determining sensor placement.

Generally, several features on an object are tracked. For *n* feature points, the Jacobian is of the form

$$\mathbf{J}_v(t) = \begin{bmatrix} \mathbf{J}_1(t) \\ \cdots \\ \mathbf{J}_n(t) \end{bmatrix} \tag{11}$$

where $\mathbf{J}_i(t)$ is the Jacobian matrix for each feature given by the 2x6 matrix in (9) or (10).The Jacobian used for vision resolvability $\mathbf{J}(\phi)$ has been rewritten as $\mathbf{J}_v(t)$ in order to distinguish between vision and force resolvability (to be derived in the next section) and to emphasize the time varying nature of resolvability.

3.1.2 Binocular Systems with Parallel Optical Axes

In this section, the Jacobian for a stereo pair with parallel optical axes observing an object described relative to a task frame is derived. The derivation is based on equations for a stereo eye-in-hand system given in (Maru et al 1994). The term *b* represents the length of the baseline of the cameras, which is the line segment between camera focal points. The origin of the camera frame lies on the baseline midway between focal points, with the -Z axis pointing towards the object task frame, as shown in Figure 4. The camera model is represented by

$$x_{Sl} = \frac{f\left(X_C + \frac{b}{2}\right)}{s_x Z_C} \qquad y_{Sl} = \frac{fY_C}{s_y Z_C} \tag{12}$$

$$x_{Sr} = \frac{f\left(X_C - \frac{b}{2}\right)}{s_x Z_C} \qquad y_{Sr} = \frac{fY_C}{s_y Z_C} \tag{13}$$

where b is the length of the baseline of the cameras, and it is assumed that f, s_x, and s_y are the same for both cameras.Through a similar derivation as in Section 3.1.1, the mapping from task space velocity to sensor space velocity can be written as

$$\begin{bmatrix} \dot{x}_{Sl} \\ \dot{y}_{Sl} \\ \dot{x}_{Sr} \\ \dot{y}_{Sr} \end{bmatrix} = \begin{bmatrix} \frac{f}{s_x Z_C} & 0 & -\frac{f\left(X_C+\frac{b}{2}\right)}{s_x Z_C^2} & -\frac{f\left(X_C+\frac{b}{2}\right)Y_T}{s_x Z^2{}_C} & \left[\frac{fZ_T}{s_x Z_C}+\frac{f\left(X_C+\frac{b}{2}\right)\left(X_T+\frac{b}{2}\right)}{s_x Z^2{}_C}\right] & -\frac{fY_T}{s_x Z_C} \\ 0 & \frac{f}{s_y Z_C} & -\frac{fY_C}{s_y Z_C^2} & -\left[\frac{fZ_T}{s_y Z_C}+\frac{fY_C Y_T}{s_y Z^2{}_C}\right] & \frac{fY_C\left(X_T+\frac{b}{2}\right)}{s_y Z^2{}_C} & \frac{f\left(X_T+\frac{b}{2}\right)}{s_y Z_C} \\ \frac{f}{s_x Z_C} & 0 & -\frac{f\left(X_C-\frac{b}{2}\right)}{s_x Z_C^2} & -\frac{f\left(X_C-\frac{b}{2}\right)Y_T}{s_x Z^2{}_C} & \left[\frac{fZ_T}{s_x Z_C}+\frac{f\left(X_C-\frac{b}{2}\right)\left(X_T-\frac{b}{2}\right)}{s_x Z^2{}_C}\right] & -\frac{fY_T}{s_x Z_C} \\ 0 & \frac{f}{s_y Z_C} & -\frac{fY_C}{s_y Z_C^2} & -\left[\frac{fZ_T}{s_y Z_C}+\frac{fY_C Y_T}{s_y Z^2{}_C}\right] & \frac{fY_C\left(X_T-\frac{b}{2}\right)}{s_y Z^2{}_C} & \frac{f\left(X_T-\frac{b}{2}\right)}{s_y Z_C} \end{bmatrix} \begin{bmatrix} \dot{X}_T \\ \dot{Y}_T \\ \dot{Z}_T \\ \omega_{X_T} \\ \omega_{Y_T} \\ \omega_{Z_T} \end{bmatrix} \tag{14}$$

where $d=x_{Sl}-x_{Sr}$ is the disparity of each corresponding feature point. The sensor space vector contains four terms representing the optical flow of the feature in both the left and right images.

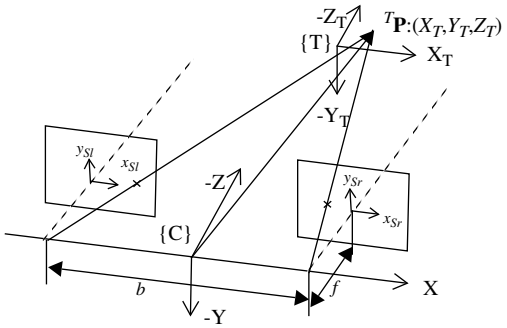

Figure 4: Task frame-camera frame definitions for a binocular system with parallel optical axes.

3.1.3 Binocular Systems with Perpendicular Optical Axes

An orthogonal stereo pair is shown in Figure 5. If the axes are aligned as shown in the figure, the Jacobian mapping from task space to sensor space can be written as

$$\begin{bmatrix} \dot{x}_{Sl} \\ \dot{y}_{Sl} \\ \dot{x}_{Sr} \\ \dot{y}_{Sr} \end{bmatrix} = \begin{bmatrix} \frac{f}{s_x Z_{Cl}} & 0 & -\frac{fX_{Cl}}{s_x Z_{Cl}^2} & -\frac{fX_{Cl}Y_T}{s_x Z_{Cl}^2} & \left[\frac{fZ_T}{s_x Z_{Cl}} + \frac{fX_{Cl}X_T}{s_x Z_{Cl}^2}\right] & -\frac{fY_T}{s_x Z_{Cl}} \\ 0 & \frac{f}{s_y Z_{Cl}} & -\frac{fY_{Cl}}{s_y Z_{Cl}^2} & -\left[\frac{fZ_T}{s_y Z_{Cl}} + \frac{fY_{Cl}Y_T}{s_y Z_{Cl}^2}\right] & \frac{fY_C X_T}{s_y Z_{Cl}^2} & \frac{fX_T}{s_y Z_{Cl}} \\ \frac{fX_{Cr}}{s_x Z_{Cr}^2} & 0 & \frac{f}{s_x Z_{Cr}} & \frac{fY_T}{s_x Z_{Cr}} & \frac{fX_{Cr}Z_T}{s_x Z_{Cr}^2} - \frac{fX_T}{s_x Z_{Cr}} & \frac{fX_{Cr}Y_T}{s_x Z_{Cr}^2} \\ \frac{fY_{Cr}}{s_y Z_{Cr}^2} & \frac{f}{s_y Z_{Cr}} & 0 & -\frac{fZ_T}{s_y Z_{Cr}} & \frac{fY_{Cr}Z_T}{s_y Z_{Cr}^2} & \frac{fY_{Cr}Y_T}{s_y Z_{Cr}^2} - \frac{fX_T}{s_y Z_{Cr}} \end{bmatrix} \begin{bmatrix} \dot{X}_T \\ \dot{Y}_T \\ \dot{Z}_T \\ \omega_{X_T} \\ \omega_{Y_T} \\ \omega_{Z_T} \end{bmatrix} \tag{15}$$

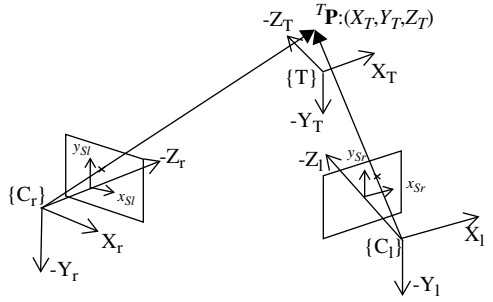

Figure 5: Task frame-camera frame definitions for a binocular system with perpendicular optical axes.

3.1.4 Vision Resolvability Ellipsoids

Vision resolvability measures the ability of a visual sensor to resolve object positions and orientations. For example, a typical single camera system has the ability to accurately resolve object locations that lie in a plane parallel to the image plane, but can less accurately resolve object depth based on the projection of object features on the image plane. Similarly, rotations within planes parallel to the image plane can be more accurately resolved than rotations in planes perpendicular to the image plane. The degree of vision resolvability is dependent on many factors. For example, depth, focal length, number of features tracked and their image plane coordinates, position and orientation of the camera, and relative positions and orientations of multiple cameras, all effect the magnitudes and directions of resolvability. Due to the difficulty in understanding the multi-dimensional nature of resolvability, we propose the vision resolvability ellipsoid as a geometrical representation of the ability of different visual sensor configurations to resolve object positions and orientations. To show the ellipsoidal representation, the Jacobian mapping is decomposed into two mappings, one representing translational components and one representing rotational components.

In Figures 6 and 7, ellipsoids for a monocular system are shown in which the two examples have the same magnification (f/Z_C), but the object is located at different depths. Figure 8 is a plot of resolvability in depth versus depth and focal length. From the plot one can observe that progressively smaller depths have progressively larger effects on resolvability in depth, while focal length tends to affect depth resolvability more linearly. In practice, depth becomes limited by the depth-of-field of the lens, and a trade-off must be made between focal length, depth, depth-of-field, and field-of-view (Nelson and Khosla 1994a). Figure 9 shows the resolvability about the optical axes versus the position at which an object is observed on the image plane. The closer the object's projection falls to the boundary of the image plane, the greater the resolvability about the optical axis.

Figure 10 shows resolvability ellipsoids for a binocular system tracking a single feature. Depth can be resolved using a single feature, but not accurately relative to directions parallel to the image plane. Figure 11 shows a plot of resolvability in depth versus baseline and depth. This plot demonstrates that reducing depth is preferable to extending the baseline to improve resolvability in depth.

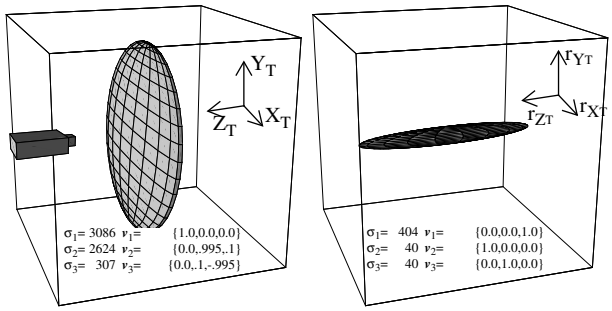

Figure 6: Resolvability Ellipsoids: monocular system, *f*=24mm, depth=1.0m, 2 features located in the task frame at (0.1m,0.1m,0) and (-0.1m,0.1m,0).

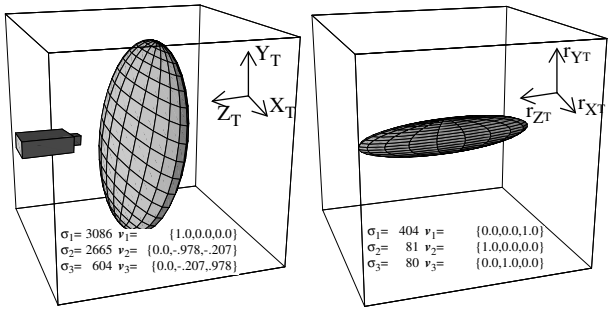

Figure 7: Resolvability Ellipsoids: monocular system, *f*=12mm, depth=0.5m, 2 features located in the task frame at (0.1m,0.1m,0) and (-0.1m,0.1m,0).

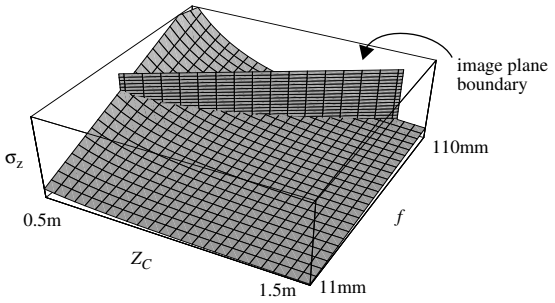

Figure 8: Resolvability of depth versus depth of object and focal length for two features located in the task frame at (0.05m,0,0) and (-0.05m,0,0).

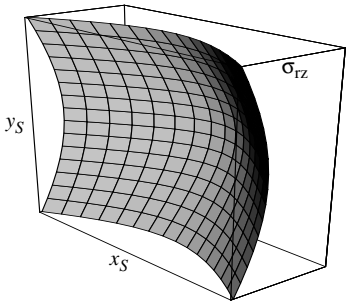

Figure 9: Resolvability in orientation about *Z* versus center of object projection onto the image plane.

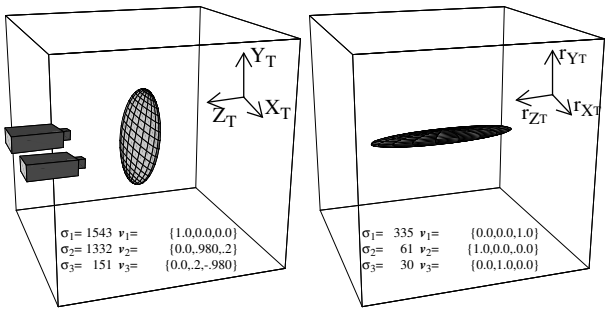

Figure 10: Resolvability Ellipsoids: stereo pair-parallel optical axes, *f*=12mm, *b*=20cm, depth=1.0m, 1 feature located in the task frame at (0,0.2m,0).

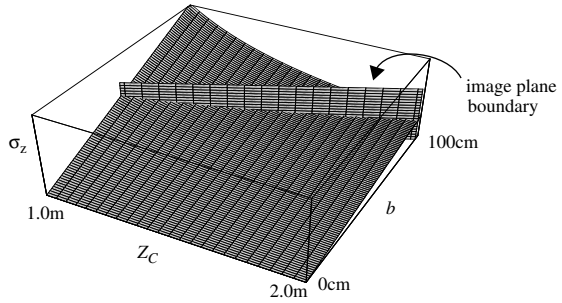

Figure 11: Resolvability in depth versus baseline length and depth of object for a stereo pair, parallel optical axes, *f*=12mm, and a single feature located at the origin of the task frame.

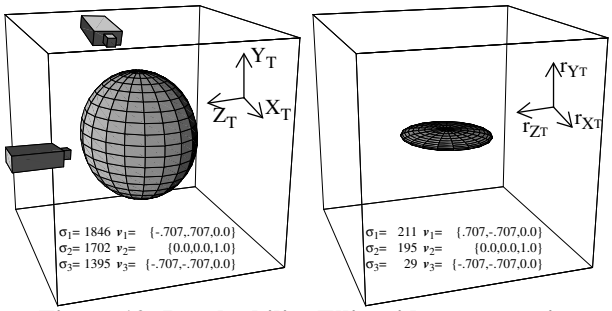

Figure 12: Resolvability Ellipsoids: stereo pair-perpendicular optical axes, *f*=12mm, depth=1.0m, 2 features located in the task frame at (-0.1m,0.1m,0), and (0.1m,-0.1m,-0.1m).

The resolvability ellipsoids for a binocular system with orthogonal optical axes are shown in Figure 12. The configuration provides a very well conditioned Jacobian mapping from task space to sensor space, although resolvability about Y_T is still relatively low.

3.2 Force Resolvability

In order to assimilate the information provided by the disparate force and vision sensors, it is necessary to develop a model of the force sensor which allows a comparison of force and vision information. The concept of sensor resolvability is used for this comparison, in which the effect of infinitesimal task space displacements are viewed in sensor space. We desire an equation of the form

$$\delta \mathbf{x}_S = \mathbf{J}_f(t)\delta \mathbf{X}_T \tag{16}$$

where $\delta\mathbf{x}_S$ is the infinitesimal displacement vector in force sensor space; $\mathbf{J}_f(t)$ is the Jacobian mapping and may be time-variable; and $\delta\mathbf{X}_T$ is the infinitesimal displacement vector in task space.

Figure 13 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. A measurement of strain $\delta\mathbf{x}_S$ taken from strain gauges mounted on the force sensor body is converted to a measurement of force $\mathbf{F}_S$ in the force sensor coordinate frame {S} through a force calibration matrix $\mathbf{C}_S$.

$$\delta\mathbf{x}_S = \mathbf{C}_S\mathbf{F}_S \tag{17}$$

$\mathbf{C}_S$ is a constant matrix that depends on the physical structure of the sensor body and the location of the strain gauges on the body. The pseudoinverse of $\mathbf{C}_S$, $\mathbf{C}_S^+$, is calculated to obtain $\mathbf{F}_S$ from measured strain gauge readings $\delta\mathbf{x}_S$ (Uchiyama et al 1991)

$$\mathbf{F}_S = \mathbf{C}_S^+\delta\mathbf{x}_S \tag{18}$$

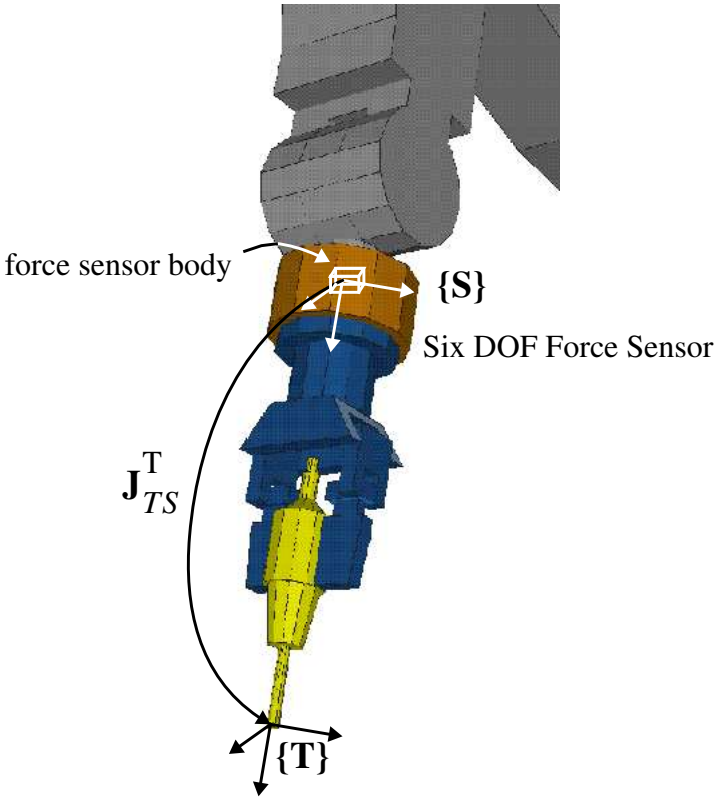

Figure 13: Coordinate frame definitions for a manipulation task that employs force sensing

$\mathbf{F}_S$ is converted to a force $\mathbf{F}_T$ in task space {T} by the Jacobian mapping of the task frame with respect to the sensor frame

$$\mathbf{F}_T = \mathbf{J}_{TS}^{\mathrm{T}}\mathbf{F}_S \tag{19}$$

Strain gauge measurements, then, are converted to forces in the task space via the equation

$$\mathbf{F}_T = \mathbf{J}_{TS}^{\mathrm{T}}\mathbf{C}_S^{+}\delta\mathbf{x}_S \tag{20}$$

During contact stages of manipulation, particular components of $\mathbf{F}_T$ are the quantities to be controlled. However, when using force and vision feedback together, force measurements are meaningless in terms of visual feedback. Therefore, we define a system stiffness $\mathbf{K}$ in order to arrive at a relationship between task displacement $\delta\mathbf{X}_T$ and task force $\mathbf{F}_T$

$$\mathbf{F}_T = \mathbf{K}\delta\mathbf{X}_T \tag{21}$$

This formula applies to quasi-static cases only, therefore inertial and damping terms are ignored. This assumption is valid since we are concerned with the resolution of the sensor, rather than its bandwidth properties.

In order to model the stiffness of the system $\mathbf{K}$, we must consider sources of compliance. We assume rigid objects are being manipulated, therefore no compliance exists in the objects. The sensor itself is obviously compliant, since it measures strain. Another important source of compliance is the manipulator itself. A rough stiffness analysis shows that the manipulator introduces the vast majority of task compliance, therefore we ignore sensor stiffness and concentrate on the compliance in the manipulator for the system stiffness model.

To analyze the relationship between end-effector stiffness and end-effector displacements, we use an augmented form of Kim's Premultiplier Diagram (Kim et al 1992), shown in Figure 14. The Premultiplier Diagram describes the static relationships between manipulator forces and positions in both joint and end-effector coordinates for redundant and non-redundant manipulators. In addition to (21), the diagram also illustrates the following relationships

$$\delta\mathbf{X}_T = \mathbf{J}_M(\theta)\delta\theta \tag{22}$$

$$\tau = \mathbf{J}_M^{\mathrm{T}}(\theta)\mathbf{F}_T \tag{23}$$

$$\tau = \mathbf{K}_\theta\delta\theta \tag{24}$$

where $\mathbf{J}_M(\theta)$ is the manipulator Jacobian matrix and varies with θ, the vector of joint positions; $\delta\theta$ is the infinitesimal displacement vector in joint space; τ is the vector of joint torques; and $\mathbf{K}_\theta$ is the joint stiffness matrix. Various vectors can be derived in terms of one another by traversing a path through the diagram and combining the proper mappings. As previously mentioned, we desire an expression for the system stiffness $\mathbf{K}$ in terms of known quantities for modeling purposes. By traversing the Premultiplier diagram from $\mathbf{F}_T$ to $\delta\mathbf{X}_T$ via the joint variables τ and $\delta\theta$, we derive the expression

$$\mathbf{F}_T = \mathbf{J}_M^{-\mathrm{T}}(\theta)\mathbf{K}_\theta\mathbf{J}_M^{-1}(\theta)\delta\mathbf{X}_T \tag{25}$$

$\mathbf{J}_M(\theta)$ is known because the kinematic structure of the manipulator is known. From the control law used to command joint torques, the joint stiffness $\mathbf{K}_\theta$ can be derived. For example, the most common strategy for controlling a manipulator is with inner loop PD (proportional-derivative) position controllers at each joint. For a joint PD control scheme operating under quasi-static assumptions, the joint stiffness is simply the value of the proportional gain (neglecting joint friction). Therefore, the system stiffness can be expressed as

$$\mathbf{K} = \mathbf{J}_M^{-\mathrm{T}}(\theta)\mathbf{K}_\theta\mathbf{J}_M^{-1}(\theta) \tag{26}$$

and depends on the configuration of the manipulator as well as the stiffness of the joint controllers.

From the augmented Premultiplier Diagram, the Jacobian mapping from task space to force sensor space is written as

$$\delta\mathbf{x}_S = \mathbf{C}_S\mathbf{J}_{ST}^{\mathrm{T}}\mathbf{J}_M^{-\mathrm{T}}(\theta)\mathbf{K}_\theta\mathbf{J}_M^{-1}(\theta)\delta\mathbf{X}_T \tag{27}$$

where

$$\mathbf{J}_f(k) = \mathbf{C}_S\mathbf{J}_{ST}^{\mathrm{T}}\mathbf{J}_M^{-\mathrm{T}}(\theta)\mathbf{K}_\theta\mathbf{J}_M^{-1}(\theta) \tag{28}$$

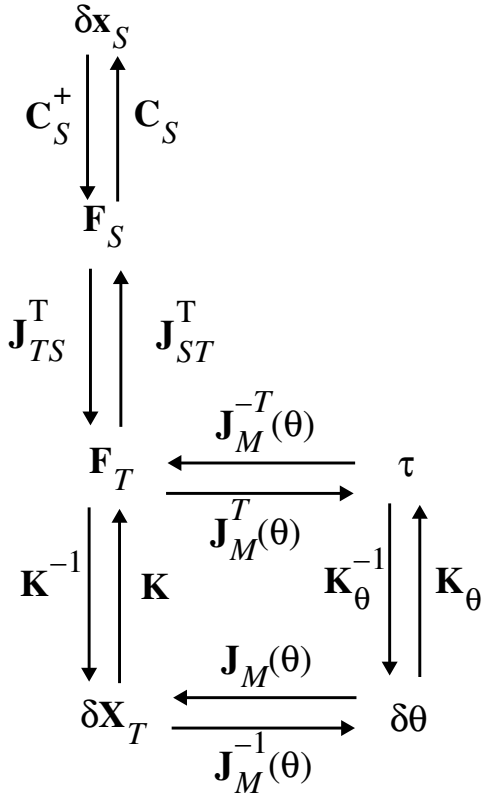

Figure 14: An augmented form of Kim's Premultiplier Diagram (Kim et al 1992) for illustrating transformations between force sensor strain gauge measurements and end-effector cartesian displacements under quasi-static conditions.

The principle components of this mapping can be used to determine the force resolvability of various sensor-manipulator-task configurations. These components are then compared with vision resolvability for assimilating information from the two disparate sensing modalities during task execution.

3.3 Comparing Vision and Force 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. 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 in Section 6, the value is typically around 1.0 pixels. This variance is translated into the task space domain through the pseudoinverse of the image Jacobian used for vision resolvability

$$\sigma_T = \mathbf{J}_v^+(k)\sigma_S \tag{29}$$

where σ_T is the vector of positional variance in task space; $\mathbf{J}_v^+(k)$ is the pseudoinverse of the image Jacobian; and σ_S is the vector representing feature variance in sensor space. For the camera-lens configuration given in Figure 6, the task positional variance is on the order of 0.0003m in a plane parallel to the image plane and 0.003m along the optical axis.

To determine force sensitivity to task space displacements, we must invert the force resolvability matrix. This is written as

$$\sigma_T = \mathbf{J}_M(\theta)\mathbf{K}_\theta^{-1}\mathbf{J}_M^{\mathrm{T}}(\theta)\mathbf{J}_{TS}^{\mathrm{T}}\mathbf{C}_S^+\sigma_S \tag{30}$$

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. This is discussed in more detail in Section 5.

As a task proceeds, the resolvability of the two sensors, force and vision, is continuously monitored. As a surface is approached, vision resolvability eventually becomes insufficient to provide meaningful control inputs to the manipulator cartesian controller. This indicates that the force sensor can now provide valid feedback on the task even though contact has not actually occurred, and force sensor information should be considered as the primary sensing modality. This means that the task model must be capable of representing geometrical relationships among objects to be mated. This model could exist in 2D image coordinates, or in a 3D world coordinate frame projected through the camera model, for example equations (1) and (2) for a monocular system.

It is important to note that at no time is the estimate of task displacement resolution used to control the task itself. The estimate is used only to compare the capabilities of each sensor. For vision resolvability, the variance of sensor noise is used as a threshold to determine when visual servoing is no longer relevant to the task. For force resolvability, the measure is used to determine the relative stiffness and force resolution of different manipulator-task configurations. The measure is also used to ensure that the resolution of the force sensor configuration provides more accurate positional feedback than the vision sensor, which is almost always the case.

From an analysis of force resolvability, it becomes evident that for a stiff manipulator very small displacements in the task frame result in relatively large measured strains in the force sensor space. If more compliant manipulator joint controllers are implemented, larger displacements in the task frame are needed in order to induce similar strains. In terms of resolvability, this means that stiff manipulators can more easily resolve task space displacements. Therefore, stiff manipulators can more accurately position objects based on force sensor readings. Of course, this is not the complete story, because stiff controllers are also much less stable in the face of modeling errors. As is the case with any control system, a trade off must be made between performance, evaluated in this case with respect to positioning accuracy, and stability.

One should realize that the force control algorithm employed will, of course, have an effect on system stiffness. However, when evaluating force resolvability the force control algorithm is not considered. This is because we determine when to switch to and from pure force control based on visual resolvability. Force resolvability is used only to ensure that the force sensor will provide more resolvable positional feedback than the visual feedback will provide. Under visual servoing control a cartesian velocity controller is used to drive the manipulator in cartesian space, therefore, the system stiffness is a result of joint PD controllers driven by end-effector velocity commands. The quasi-static assumption used for the force resolvability derivation is valid because as contact becomes imminent, the visual servoing controller reduces commanded velocities.

4. Visual Servoing Formulation

4.1 Controller

The state equation for the visual servoing system is created by discretizing (11) and rewriting the discretized equation as

$$\mathbf{x}(k+1) = \mathbf{x}(k) + T\mathbf{J}_v(k)\mathbf{u}(k) \tag{31}$$

where M is the number of features being tracked; $\mathbf{x}(k) \in R^{2M}$; T is the sampling period of the vision system; and $\mathbf{u}(k) = \left[\dot{X}_T\ \dot{Y}_T\ \dot{Z}_T\ \omega_{X_T}\ \omega_{Y_T}\ \omega_{Z_T}\right]^{\mathrm{T}}$, the manipulator end-effector velocity. The camera Jacobian $\mathbf{J}_v(k)$ for the experimental system varies with time due to the changing feature coordinates on the image plane. The intrinsic parameters of the experimental camera-lens system are constant. The variable for time, t, is discretized at each time instant to kT, and we let $k=kT$ to simplify our equations without loss of generality.

A control strategy can be derived using the controlled active vision paradigm (Papanikolopoulos, Khosla and Kanade 1991). The control objective of the visual tracking system is to control end-effector motion in order to place the image plane coordinates of features on the target at some desired position. The desired image plane coordinates could be constant or changing with time. The control strategy used to achieve the control objective is based on the minimization of an objective function at each time instant. The objective function places a cost on differences in feature positions from desired positions, as well as a cost on providing control input, and is of the form

$$F(k+1) = [\mathbf{x}(k+1) - \mathbf{x}_D(k+1)]^T\mathbf{Q}[\mathbf{x}(k+1) - \mathbf{x}_D(k+1)] + \boldsymbol{u}^T(k)\mathbf{L}\mathbf{u}(k) \tag{32}$$

This expression is minimized with respect to the current control input $\mathbf{u}(k)$. The end result yields the following expression for the control input

$$\mathbf{u}(k) = -(\mathbf{J}_v^{\mathrm{T}}(k)\mathbf{Q}\mathbf{J}_v(k) + \mathbf{L})^{-1}\mathbf{J}_v^{\mathrm{T}}(k)\mathbf{Q}[\mathbf{x}(k) - \mathbf{x}_D(k+1)] \tag{33}$$

where $T\mathbf{J}_v(k)$ is rewritten as $\mathbf{J}_v(k)$ without loss of generality. The weighting matrices $\mathbf{Q}$ and $\mathbf{L}$ allow the user to place more or less emphasis on the feature error and the control input. Their selection effects the stability and response of the tracking system. The $\mathbf{Q}$ matrix must be positive semi-definite, and $\mathbf{L}$ must be positive definite for a bounded response. Although no standard procedure exists for choosing the elements of $\mathbf{Q}$ and $\mathbf{L}$, general guidelines can be found in (Papanikolopoulos, Nelson and Khosla 1992).

The system model and control derivation can be extended to account for system delays, modeling and control inaccuracies, and measurement noise. See (Papanikolopoulos, Nelson and Khosla 1992) for a detailed explanation of how this can be accomplished.

4.2 Feature Tracking

The measurement of the motion of the features on the image plane must be done continuously and quickly. The method used to measure this motion is based on optical flow techniques and is a modification of the method proposed in (Anandan 1987). 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. The displacement of a point $\mathbf{p}_a=(x,y)$ at the next time increment to $\mathbf{p}_{a'}=(x+\Delta x, y+\Delta y)$, is determined by finding the displacement $\Delta\boldsymbol{x}=(\Delta x,\Delta y)$ which minimizes the SSD measure

$$e(\mathbf{p}_a, \Delta\mathbf{x}) = \sum_W [I_a(x+i, y+j) - I_{a'}(x+i+\Delta x, y+j+\Delta y)]^2 \quad (34)$$

where I_a and $I_{a'}$ are the intensity functions from two successive images and W is the window centered about the feature point which makes up the feature template. For the algorithm implemented, W is 16x16 pixels, and possible displacements of up to $\Delta x=\Delta y=\pm 32$ pixels are considered. Features on the object that are to be tracked can be selected by the user, or a feature selecting algorithm can be invoked. Features with strong intensity gradients in perpendicular directions, such as corners, are typically the best features to select.

In order to decrease the search space, a pyramidal search scheme shown in Figure 15 has been implemented which first searches a coarse resolution of the image that has 1/16 the area of the original image, using a feature template in which a W that is originally 32x32 is averaged to 8x8. After determining where the feature is in the coarse image, a finer resolution image that is 1/4 the original spatial resolution is searched with an original W of 16x16 which is averaged to 8x8 in an area centered about the location of the minimum SSD measure found in the coarse image. Finally, the full resolution image and the 16x16 feature template are used to pinpoint the location of the displaced feature.

The pyramidal scheme reduces the time required for the computation of the SSD algorithm by a factor of five for a single feature over the method of computing the feature locations at the full resolution alone. However, reliability can be sacrificed when the selected feature loses its tracking properties (strong perpendicular intensity gradients) at the coarser image resolutions. Since the search scheme first estimates where the feature is located based on the coarse image, it is critical that good features at coarse resolutions are tracked. When a user selects features, it is often not obvious that a particular feature may lose its tracking characteristics at coarse resolutions. Because of this, an automatic feature selector has been implemented based on (Tomasi and Kanade 1991) which accounts for the different levels of resolution in the pyramidal search scheme.

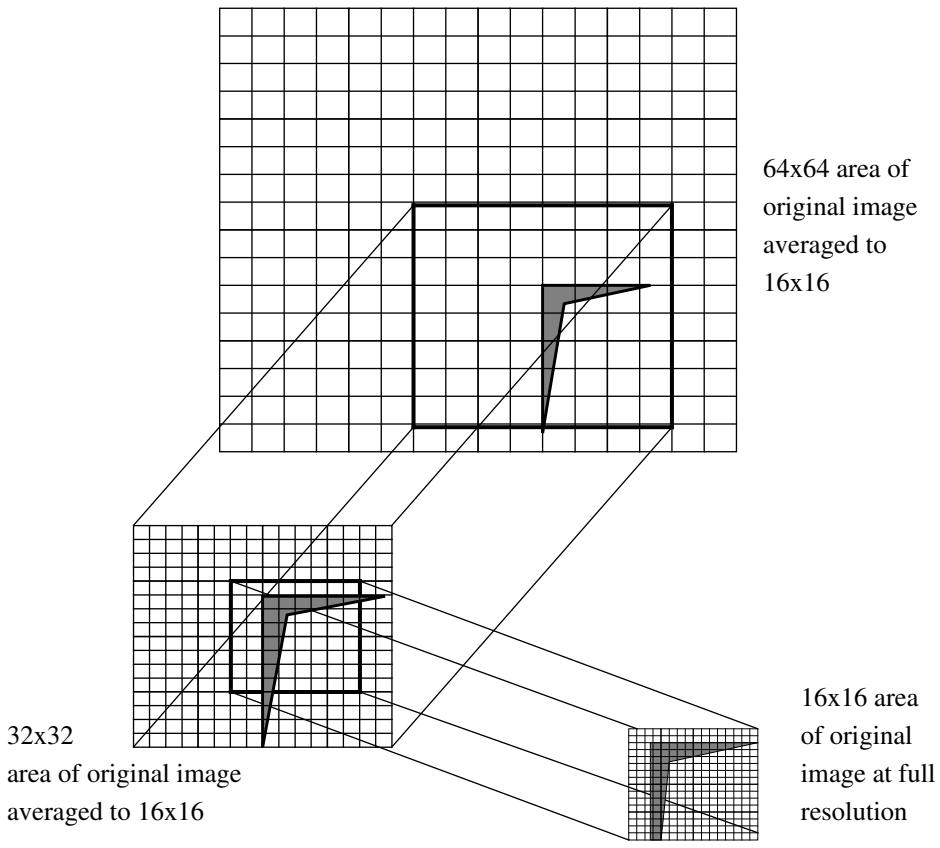

Figure 15: A pyramidal search scheme is used in the SSD optical flow algorithm in order to increase the overall sampling rate of the system.

5. Vision/Force Servoing

In order to illustrate the advantages of assimilating disparate sensor feedback using our proposed method, we experimentally demonstrate the performance of the technique during contact transitions. To quickly and efficiently perform robotic manipulation tasks in uncertain environments, a robotic end-effector must be able to successfully approach and contact objects rapidly using sensor feedback. In a rigid environment this is difficult. With a stiff manipulator this becomes even more difficult because neither the surface nor the manipulator are able to easily dissipate excess energy upon contact. However, the most common form of force control uses a proportional-derivative strategy with high proportional gains and low damping, which is an inherently stiff system. The reason this type of force control is popular is because it is simple to implement, choosing gains is easy, and it achieves a relatively high bandwidth once contact is successfully made. The problem with this strategy, however, is in achieving initial contact quickly and stably while maintaining low impact forces. Many researchers have studied this problem, and various impact strategies have been proposed, as discussed in Section 2.1. However, the fundamental problem of using force feedback alone to minimize impact forces while quickly achieving contact stably within imprecisely calibrated environments still exists. By combining vision feedback with force feedback using the concept of resolvability in a nonlinear control strategy, we demonstrate how fast stable contact transitions with a stiff manipulator in a rigid environment can be achieved.

The force control portion of our proposed visual/force servoing strategies is based on past work in hybrid force control. The implemented force control scheme is a combination of hybrid force/position control (Raibert and Craig 1981) and damping force control (Whitney 1977), resulting in a hybrid force/velocity control scheme. Because the dynamics, particularly friction, of the laboratory robot (a Puma 560) are difficult to accurately model, a simple cartesian control scheme is used in which a manipulator Jacobian inverse converts cartesian velocities to joint velocities, which are then integrated to joint reference positions. High servo rate (500Hz) PD controllers are implemented for each joint in order to follow joint trajectories which achieve the desired cartesian motion.

If simple force damping control is used to impact surfaces, a manipulator can easily become unstable unless force gains are tuned to extremely low values resulting in unacceptably slow motion during the approach phase of the task. Because of this, most manipulation strategies use a guarded move to initiate contact with a surface. During a guarded move, surfaces are approached under position control while the force sensor is monitored. If the sensed force exceeds a threshold, motion is immediately stopped and a force control strategy can then be invoked. The main limitation of this strategy is that high contact forces can result unless the effective mass of the manipulator is low so that the end-effector can be quickly stopped before contact forces increase significantly.

The proper use of visual feedback can overcome the problems exhibited by guarded move and pure force control strategies upon impact. Visual servoing improves manipulator performance during contact transitions by incorporating information regarding the proximity of the surface to be con-

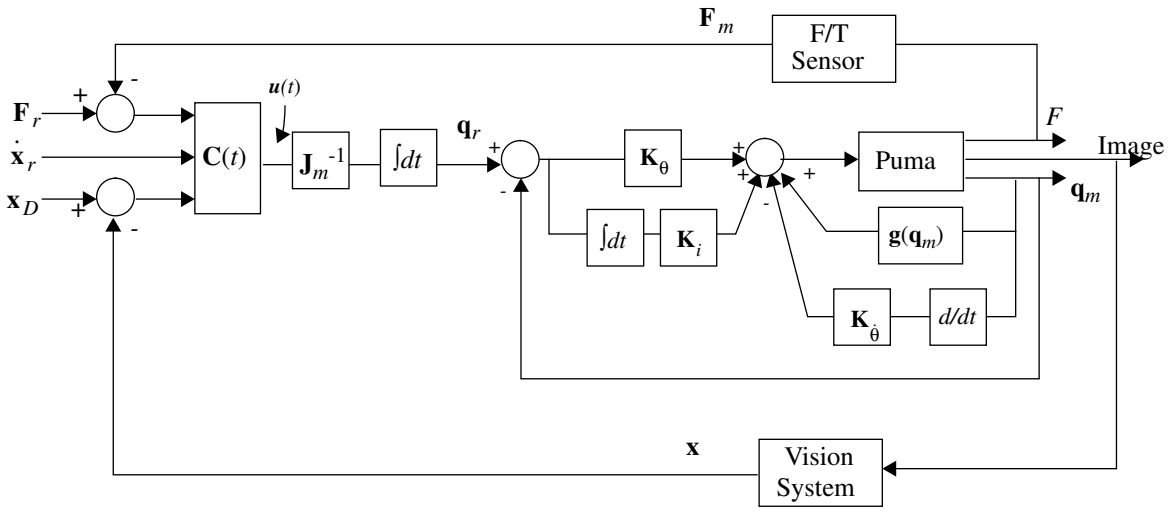

Figure 16: Force and vision in the feedback loop.

tacted in the manipulator feedback loop. When the end-effector is far from a surface, visual servoing commands fast end-effector motion. The speed of the approach decreases as the end-effector approaches nearer the surface. Contact can then be stably initiated through the use of low gain force controllers. A generic control framework for visual/force servoing is shown in Figure 16.

A fundamental problem when sharing control between force and vision sensors occurs due to end-effector inertial effects. Because force sensors measure all forces (inertial, gravitational, and tactile), the inertial coupling of the end-effector mass beyond the sensor introduces inertial forces into force sensor readings. When the vision system commands motions, the resulting accelerations cause unstable excitations of the force control system. In order to compensate for the unstable excitations, it is necessary to develop robust strategies for avoiding the excitations. Thresholding of force readings is not feasible, since inertial effects can often be as large as desired contact forces. Figure 17 shows the magnitude of experimentally determined inertial forces and the associated measured cartesian accelerations which cause these forces.

We have developed a robust vision/force control strategy based on the fact that large accelerations induce inertial forces. If visual servoing results in measurable end-effector accelerations of sufficient magnitude, then force readings in directions opposite to these accelerations are being induced. Because measured cartesian accelerations are derived from joint encoder readings, thus requiring two differentiations of measured joint values and a transformation from joint space to cartesian space, measured cartesian accelerations are noisy. Therefore, we also consider the measured direction of end-effector motion. If measured cartesian accelerations have been induced by

visual servoing, and if a measurable cartesian velocity exists, then sensed forces must be due to inertial coupling, and force control commands should be ignored. This strategy can be written as

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

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

for each axis, i {

if $\quad ((\,(|\ddot{x}_{m_i}| > \varepsilon_a) \wedge (\dot{x}_{m_i} \mathrm{sgn} F_{m_{i_i}} < \varepsilon_v)\,) \vee (\dot{x}_{ref_{vi}} F_{m_i} > 0.0) \vee (|F_{m_i}| < F_{Th})\,)$

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

else

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

}

$$\mathbf{u}(k) = \mathbf{S}_v\dot{\mathbf{x}}_{ref_v} + \mathbf{S}_r\dot{\mathbf{x}}_r + \mathbf{S}_F\dot{\mathbf{x}}_{ref_f} \tag{35}$$

where $\mathbf{x}$ is the feature vector representing the object being servoed; $\mathbf{x}_D$ represents a state in feature space that will bring the object being servoed into contact with some surface; $\mathbf{S}_F$ is the matrix that selects axes along which force control will be applied; $\mathbf{G}_F$ is the matrix of force control gains; $\mathbf{F}_r$ and $\mathbf{F}_m$ represent reference and measured forces with respect to the task coordinate frame {T}; $\dot{x}_{m_i}$ and $\ddot{x}_{m_i}$ represent measured cartesian velocities and accelerations of the end-effector in task space; $\dot{\mathbf{x}}_r$ is some desired reference end-effector velocity due, for example, to trackball input from a teleoperator; $\mathbf{S}_r$ is the matrix that selects axes along which this input will be applied; and ε_a, ε_v, and F_{Th} threshold sensor noise.

For teleoperation tasks guided by visual servoing, compliant contact with the environment will occur if (35) is used alone, assuming the teleoperator adjusts the desired visual feature state to be at or below the surface to be contacted. For autonomous manipulation, however, this strategy does not ensure contact will occur if the actual location of the surface is beyond the visual estimate of the surface. During autonomous manipulation, the strategy given by (35) must be rewritten as

$$\mathbf{u}(k) = \begin{cases} (35)\,, & \|\mathbf{x}_D(k) - \mathbf{x}(k)\| > \varepsilon \quad (36)\,\text{a} \\ \mathbf{S}_F\mathbf{G}_F\,(\mathbf{F}_r(k) - \mathbf{F}_m(k))\,, & \|\mathbf{x}_D(k) - \mathbf{x}(k)\| \le \varepsilon \quad (36)\,\text{b} \end{cases}$$

in order to ensure that contact will occur. Manipulator motion is first controlled by the strategy given in (35). The controller then switches to pure force control if the error between desired and measured visual feature states converges to within a threshold. This threshold is derived from the variance of the noise in the vision sensor,

$$\varepsilon = 2.0\|\dot{\sigma}_S\| \tag{37}$$

where σ_S is the feature variance vector on the image plane. Stable impact with a surface can then be achieved, large contact forces can be minimized, and bounce can be avoided.

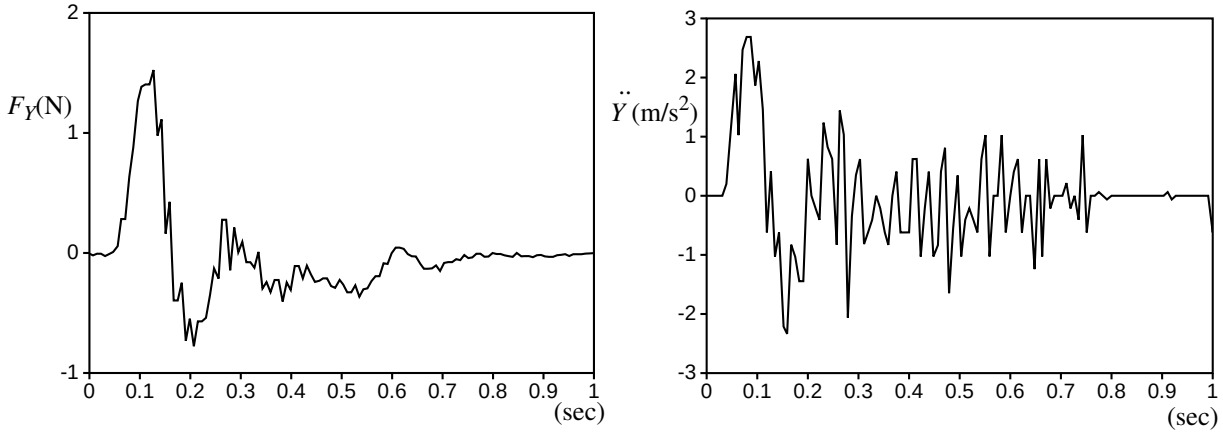

Figure 17: Inertial forces measured by the force sensor and the corresponding measured cartesian accelerations which induced these forces.

Figure 18: Laboratory setup used for performing vision/force servoing experiments.

6. Experimental Results

6.1 Hardware Setup

The vision/force servoing algorithms previously described have been implemented on a robotic assembly system consisting of three Puma 560's called the Troikabot. The Pumas are controlled from a VME bus with two Ironics IV-3230 (68030 CPU) processors, an IV-3220 (68020 CPU) processor which also communicates with a trackball, a Mercury floating point processor, and a Xycom parallel I/O board communicating with three Lord force sensors mounted on the Pumas' wrists. All processors on the controller VME run the Chimera 3.0 reconfigurable real-time operating system (Stewart, Schmitz and Khosla 1992). An Adept robot is also used for providing accurate target motion. The entire system is shown in Figure 18.

A diagram of the hardware setup is shown in Figure 19. The vision system VME communicates with the controller VME using BIT3 VME-to-VME adapters. The Datacube Maxtower Vision System calculates the optical flow of the features using the SSD algorithm discussed in Section 4.2. A special high performance floating point processor on the Datacube is used to calculate the optical flow of features, and a 68030 board, also on the vision system, computes the control input. An image can be grabbed and displacements for up to five 16x16 features in the scene can be determined at 30Hz. A Lord Model 15/50 force sensor provides force and torque values for each cartesian axis at 100Hz.

6.2 Results

Throughout this section, experimental results given will be referenced to the coordinate frames shown in Figure 20. For the initial set of experiments, the results of three trials are shown in which the desired goal position for the visual servoing strategy is purposely chosen to have differing magnitudes of error with respect to the true location of the surface. A final contact force of -2N is desired. This allows us to evaluate the ability of our force/vision control strategy (36) to operate under conditions in which force and vision information significantly disagree. Figure 21 shows the motion of the end-effector on the image plane for the three trials. For trials 2 and 3 the desired image plane position of the end-effector y_D actually falls beneath the true surface. In trial 2 the error in surface position is 15 pixels, and in trial 3 the error is 45 pixels. For trial 1 the estimate of the surface and the true location are in close agreement, as would normally be the case.

In trials 2 and 3, the end-effector impacts the surface after approximately 0.3s, when motion of the end-effector on the image plane abruptly stops. For trial 1, the surface is not contacted until after approximately 0.5s, because the manipulator purposely slows down before impact. The force plot in Figure 22 shows that this results in significantly reduced impact forces, and a much quicker transition to the desired contact force of -2N. When visual feedback incorrectly estimates the location

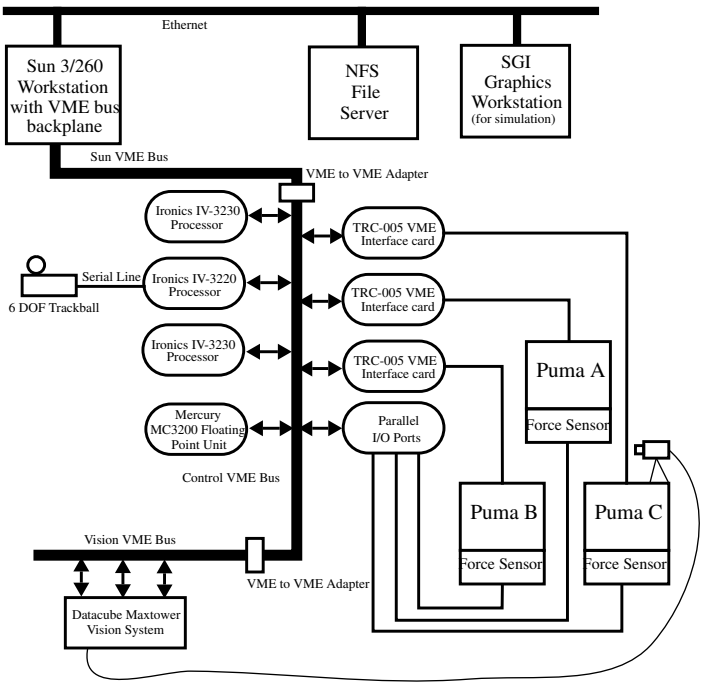

Figure 19: The Troikabot system architecture.

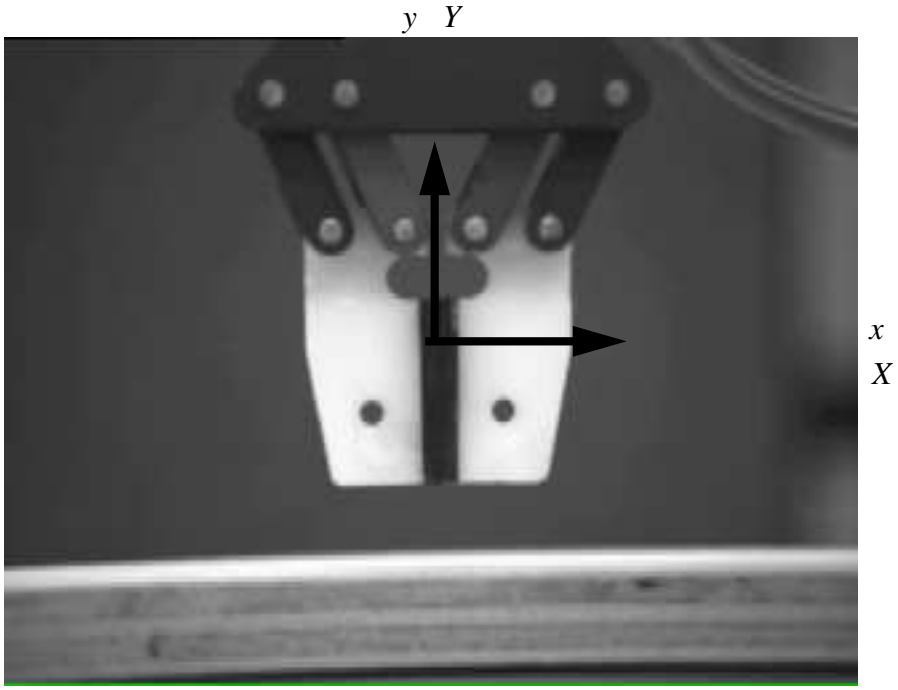

Figure 20: The camera view for visual servoing and coordinate axes on the image plane and in the task frame.

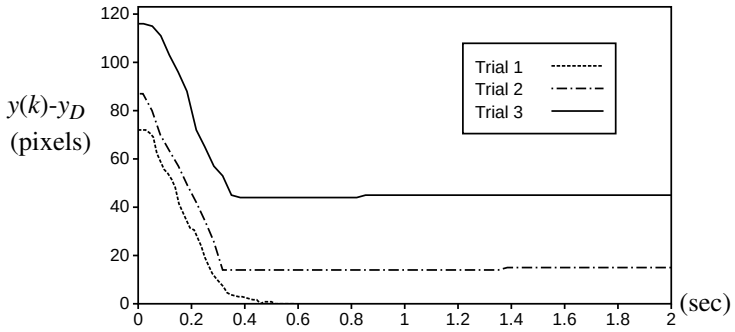

Figure 21: Vertical error between desired and measured end-effector position on the image plane for three different trials, each with a different error in the estimated location of the surface.

of the surface, as in the case of the second and third trials, high contact forces occur. If the error in the estimate falls within the surface, as in trials 2 and 3, then the poorer the estimate of the surface, the higher the contact force because the higher the commanded visual servoing velocity at impact. If the error in the surface location estimate is in the other direction, then the time in which it takes to initiate contact would increase directly with the magnitude of the error. The impact force, however, would be on the order of trial 1's impact force.

The commanded end-effector velocity for all three trials is shown in Figure 23. The solid lines correspond to $\mathbf{u}(k)$ in (35), the dashed lines to the visual servoing velocity $\dot{x}_{ref_v}$, and the dotted/dashed lines to the force servoing velocity $\dot{x}_{ref_f}$. Visual servoing brings the end-effector quickly towards the surface, and upon contact force servoing takes over. From the force plot in Figure 22, one can observe measurable inertial forces before contact actually occurs. These forces are of a magnitude greater than 1.5N, however, our proposed control strategy (35) successfully rejects these observed forces because they are not the result of contact. From Figure 23, one can see that end-effector velocities have been clipped at 0.10m/s. This is because the feature tracker can only track objects with a limited optical flow. Thus, the trial in which the surface location is in error of 45 pixels represents the worst case impact force, because the manipulator is traveling at approximately 0.10m/s at the time of impact. For these experimental results, force gains of 0.001 (m/s)/N were used, the diagonal elements of $\mathbf{Q}$ were chosen to be $2.0\text{x}10^{-6}$, and the diagonal elements of $\mathbf{L}$ were chosen to be 10.0. Thresholds were chosen to be ε_a=0.01m/s^2, ε_v=0.001m/s, and F_{Th}=1.5N.

A second set of experimental results was collected in order to illustrate the advantages of our proposed force/vision strategy over two other common impact strategies, the guarded move and pure force control. Figure 24 shows results in which our proposed force/vision servoing algorithm (36) is used to servo the end-effector to a surface 5.9cm from the initial end-effector position. A force of -2N between the end-effector and the surface 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, the manipulator travels 5.9cm in 3.1s before reaching the surface. As

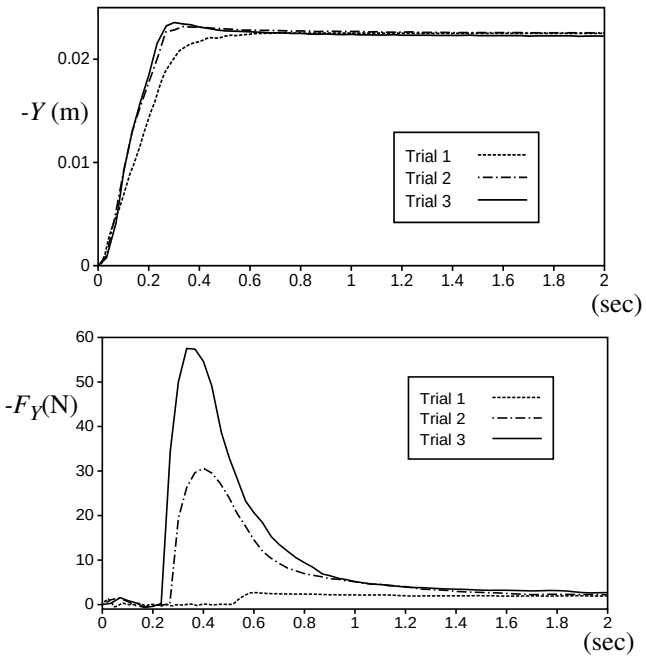

Figure 22: Vertical position of end-effector in cartesian space and force measured along the vertical direction versus time for all three trials.

soon as the surface is contacted, the manipulator becomes unstable, as Figure 25 shows. 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 26 shows a force plot of a guarded move in which the force sensor is monitored at 100Hz. High contact 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.

Figure 27 shows a comparison of the motion and force time histories for the three impact strategies. The gains used in force/vision control strategy are the same as the gains used in the previous set of experiments, including the force gain of (0.001 (m/s)/N). For force control alone, higher force gains (0.005 (m/s)/N) had to be chosen in order to induce end-effector motion of a reasonable speed in free space, but this gain, while resulting in less than half the speed of visual servoing, still proved to be highly unstable. The guarded move strategy also allowed only moderate speeds (0.02m/s) and still resulted in high impact forces. At higher speeds, extremely high impact forces would result which could have easily damaged the manipulator. Using visual servoing to bring the manipulator near the surface provides a simple technique for slowing the end-effector before contact is imminent. These results clearly show that visual servoing greatly simplifies the impact problem by providing low-level feedback on the proximity of the surface to the end-effector. The result is a fast approach velocity that generates low impact forces with no bounce.

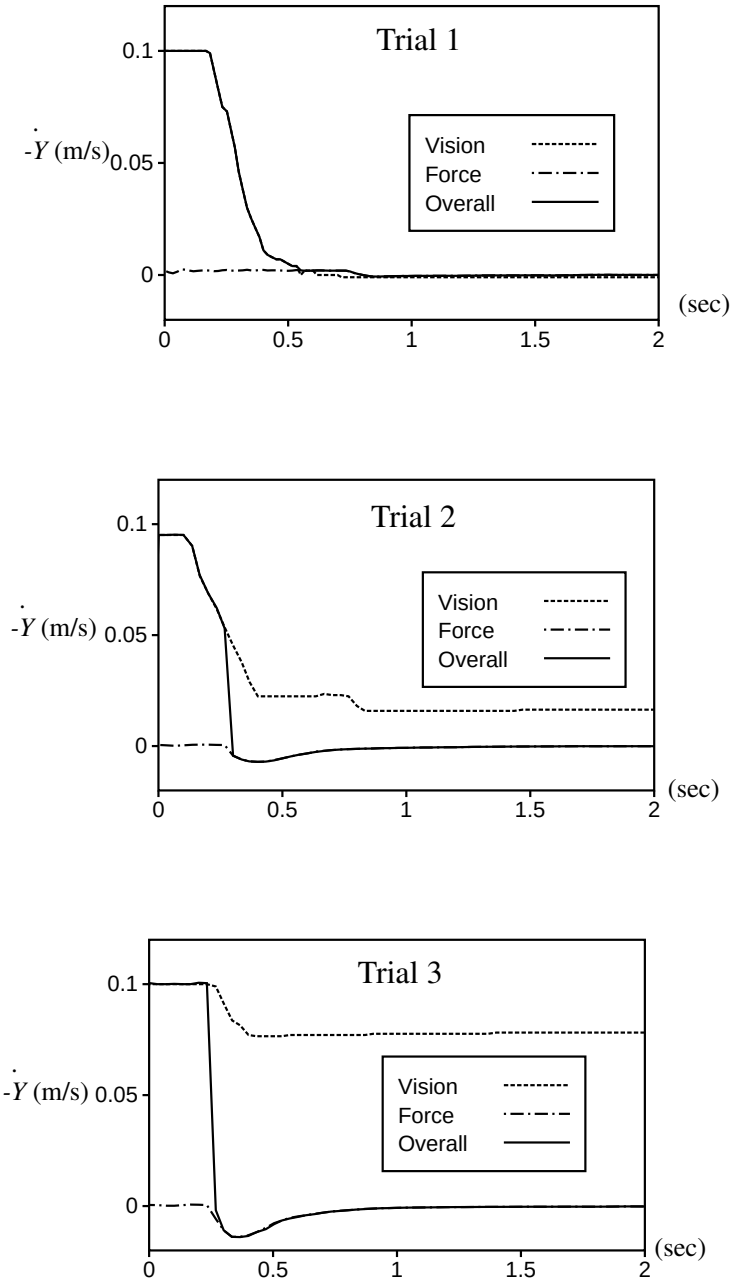

Figure 23: Commanded end-effector cartesian velocity along -*Y* for the three trials with varying error magnitudes in the estimated surface location. "Vision" corresponds to $\dot{x}_{refv}$, "Force" corresponds to $\dot{x}_{reff}$, and "Overall" corresponds to $u_y(t)$, in (35).

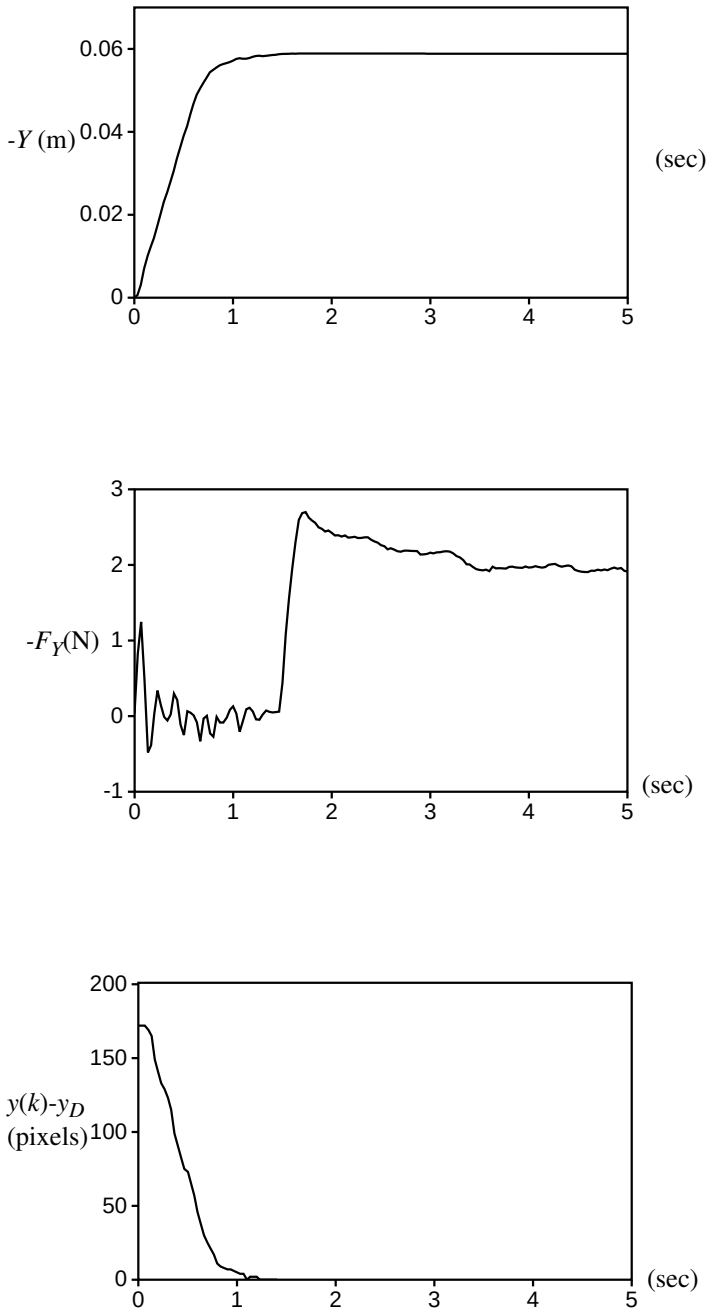

Figure 24: Vertical position of end-effector in cartesian space, force measured along the vertical direction, and pixel error versus time for traded control.

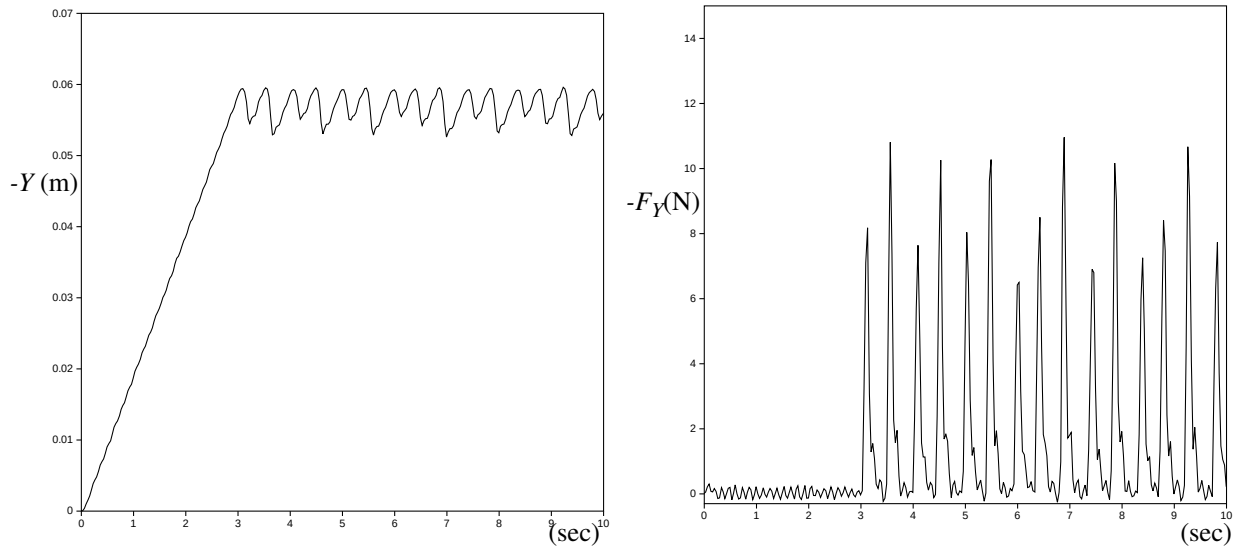

Figure 25: Vertical position of end-effector in cartesian space and force measured along the vertical direction versus time for simple damping force control upon impact with a surface.

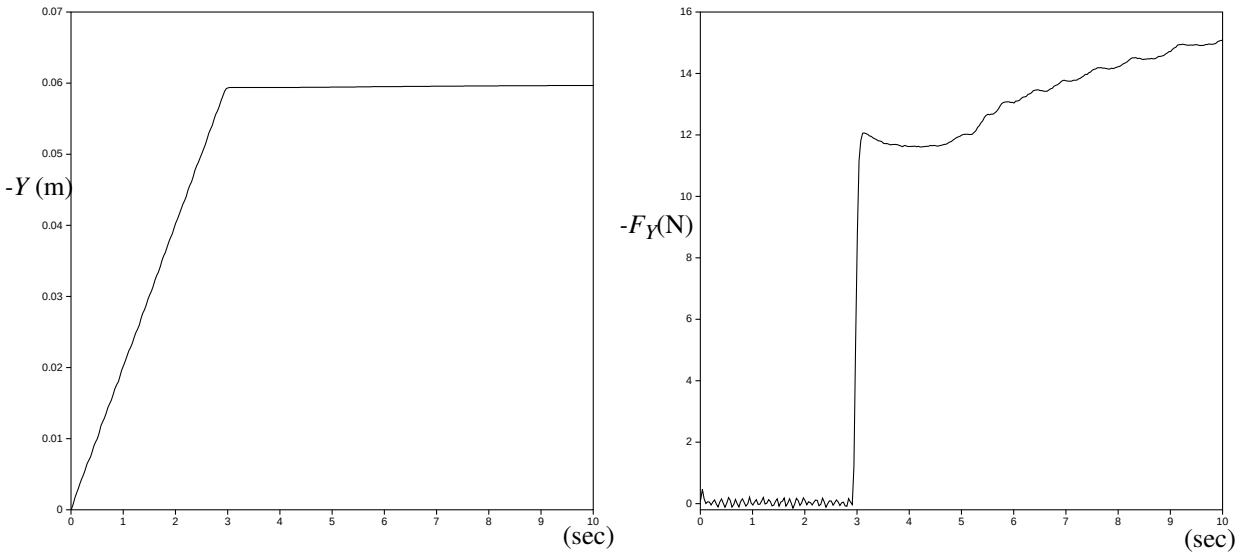

Figure 26: Vertical position of end-effector in cartesian space and force measured along the vertical direction versus time for a guarded move impact strategy that switches to position control.

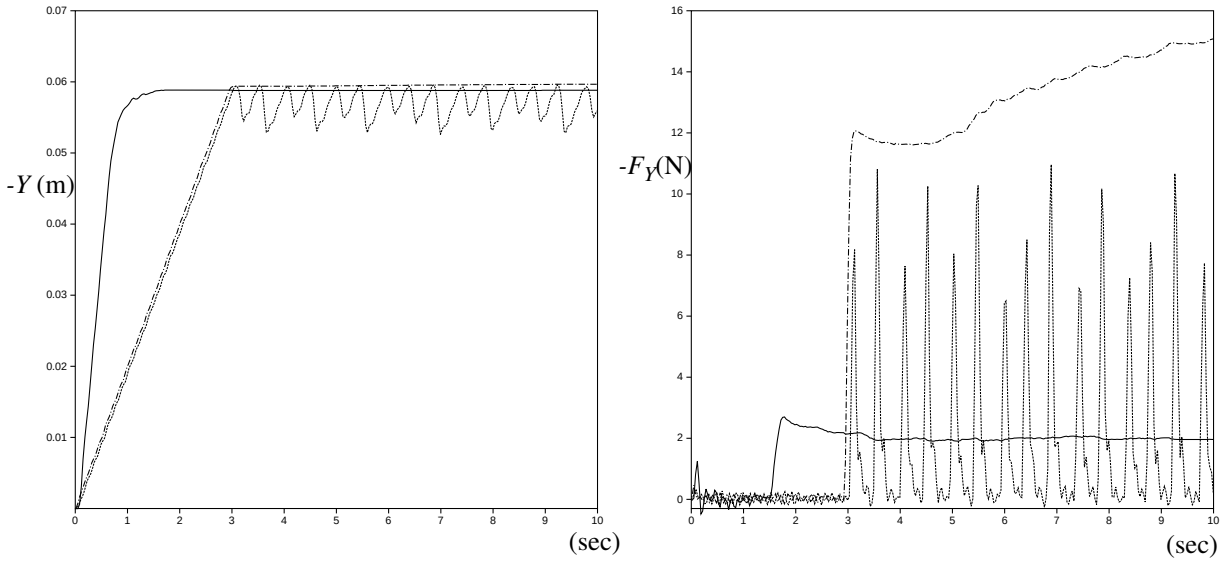

Figure 27: Combined plots of vertical position and measured force during impact for force/vision control, damping force control, and a guarded move.

7. Conclusion

Force and vision sensors provide complementary information, yet they are fundamentally different sensing modalities. This implies that traditional sensor integration techniques that require common data representations are not appropriate for combining the feedback from these two disparate sensor. In this paper, we have introduced 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 robotic manipulation tasks. By monitoring the resolvability of the two sensing modes with respect to the task, the information provided by the disparate sensors can be seamlessly assimilated during task execution. A nonlinear force/vision servoing algorithm that uses force and vision resolvability 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. References

[1] Allen, P.K. 1988. Integrating Vision and Touch for Object Recognition Tasks. *Int. J. of Robotics Research*. 7(6):15-33.

[2] Allen, P.K. 1989. Real-time motion tracking using spatio-temporal filters. *Proc. Image Understanding Workshop*. San Mateo, CA:Morgan Kaufmann, pp. 695-701.

[3] An, C. and Hollerbach, J. 1987. Dynamic stability issues in force control of manipulators. *Proc. 1987 Int. Conf. on Robotics and Automation*. New York:IEEE, pp.860-896.

[4] Anandan, P. 1987. Measuring visual motion from image sequences. Tech. Rept. COINS-TR-87-21, Amherst, Mass.:University of Massachusetts COINS Department.

[5] Castaño, A. and Hutchinson, S. 1994. Visual compliance: task-directed visual servo control. *IEEE Trans. on Robotics and Automation*. 10(3):334-342.

[6] Corke, P.I. and Paul, R.P. 1989. Video-rate visual servoing for robots. *Lecture notes in control and information science*. eds. V. Hayward and O. Khatib. London: Springer-Verlag, pp. 429-451.

[7] Corke, P.I. Visual control of robot manipulators-a review. *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, ed. K. Hashimoto. London:World Scientific, pp. 1-31.

[8] Cowan, C.K. and P.D. Kovesi, "Automatic Sensor Placement from Vision Task Requirements," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5), 407-416, 1988.

[9] Das, S. and N. Ahuja, "A Comparative Study of Stereo, Vergence, and Focus as Depth Cues for Active Vision," *Proc. of the 1992 Computer Vision and Pattern Recognition Conference (CVPR-92)*, 194-199, 1993.

[10] Durrant-Whyte, H.F. 1987. Consistent Integration and Propagation of Disparate Sensor Observation. *Int. J. of Robotics Research*. 6(3):3-24.

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

[12] Eppinger, S. and Seering, W. 1987. Understanding bandwidth limitations on robot force control. *Proc. 1987 Int. Conf. on Robotics and Automation*. New York:IEEE, pp.904-909.

[13] Espiau, B., Chaumette, F. and Rives, P. 1992. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*. 8(3):313-326.

[14] Feddema, J.T. and Lee, C.S.G. 1990. Adaptive image feature prediction and control for visual tracking with a hand-eye coordinated camera. *IEEE Trans. on Systems, Man, and Cybernetics*, 20(5):1172-1183.

[15] Ghosh, B.K., Jankovic, M. and Wu, Y.T. 1992. Some problems in perspective system theory and its application to machine vision. *Proc. 1992 IEEE/RSJ Int. Conf. on Intelligent Robots and Sys. (IROS92)*. New York:IEEE, pp. 139-146.

[16] Hager, G.D. *Task-Directed Sensor Fusion and Planning - A Computational Approach*, Kluwer Academic Publishers, 1990.

[17] Hager, G.D. Robot feedback control based on stereo vision: towards calibration-free hand-eye coordination. *Proc. 1994 Int. Conf. on Robotics and Automation*. New York:IEEE, pp. 2850-2856.

[18] Hashimoto, K. and Kimura, H. 1993. LQ optimal and nonlinear approaches to visual servoing. *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, ed. K. Hashimoto. London:World Scientific, pp. 165-198.

[19] Hogan, N. 1985. Impedance control: an approach to manipulation, Parts I-III. *ASME J. Dyn. Sys., Measurements, and Control*. 107(1):1-24.

[20] Hogan, N. 1987. Stable execution of contact tasks using impedance control. *Proc. 1987 Int. Conf. on Robotics and Automation*. New York:IEEE, pp.1047-1054.

[21] Hosoda, K. and Asada, M. 1994. Versatile visual servoing without knowledge of true jacobian. *Proc. 1994 IEEE/RSJ Int. Conf. on Intelligent Robots and Sys. (IROS94)*. New York:IEEE, to appear.

[22] Hyde, J.M. and Cutkosky, M.R. 1993. Contact transition: an experimental study. *Proc. 1993 Int. Conf. on Robotics and Automation*. New York:IEEE, pp.363-368.

[23] Ishikawa, J., Kosuge, K. and Furuta, K. 1990. Intelligent control of assembling robot using vision sensor. *Proc. 1985 Int. Conf. on Robotics and Automation*. New York:IEEE, pp. 1904-1909.

[24] Jain, R. 1989. Environment Models and Information Assimilation. Tech. Rept. RJ 6866 (65692) IBM-Yorktown Heights.

[25] Kazerooni, H. 1987. Robust, non-linear impedance control for robot manipulators. *Proc. 1987 Int. Conf. on Robotics and Automation*. New York:IEEE, pp.741-750.

[26] Khatib, O. and Burdick, J. 1986. Motion and force control for robot manipulators. *Proc. 1986 Int. Conf. on Robotics and Automation*. New York:IEEE, pp.1381-1386.

[27] Kim, J.O., P. Khosla, and W.K. Chung, "Static Modeling and Control of Redundant Manipulators," *Robotics and Computer-Integrated Manufacturing*, 9(2), 145-157, 1992.

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

[29] Koivo, A.J. and Houshangi, N. 1991. Real-time vision feedback for servoing of a robotic manipulator with self-tuning controller. *IEEE Trans. on Sys, Man, and Cybernetics,* 21(1):134-142.

[30] Maru, N., H. Kase, S. Yamada, A. Nishikawa, and F. Miyazaki, "Manipulator Control by Visual Servoing with the Stereo Vision," *Proc. of the 1993 IEEE/RSJ Int. Conf. on Intelligent Robots and System (IROS-93)*, 1866-1870, 1993.

[31] Nakamura, Y., T. Yoshikawa, and I. Futamata, "Design and Signal Processing of Six-Axis Force Sensors," *Robotics Research 4*, eds. R. Bolles and B. Roth, 75-81, MIT Press, Cambridge, MA, 1988.

[32] Nelson, B., N.P. Papanikolopoulos, and P.K. Khosla. 1993. Visual servoing for robotic assembly. *Visual Servoing—Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*. ed. K. Hashimoto. River Edge, NJ:World Scientific Publishing Co. Pte. Ltd. pp. 139-164.

[33] Nelson, B. and Khosla, P.K. 1994a. Integrating Sensor Placement and Visual Tracking Strategies. *EXPERIMENTAL ROBOTICS III: The Third International Symposium*, *Kyoto, October 28-30 1993*, eds. T. Yoshikawa and F. Miyazaki, Springer-Verlag:London, 169-181.

[34] Nelson, B.J. and Khosla, P.K. 1994b. The Resolvability Ellipsoid for Visual Servoing. *Proc. of the 1994 Conf. on Computer Vision and Pattern Recognition (CVPR94)*, 829-832.

[35] Papanikolopoulos, N.P., Khosla, P.K. and Kanade, T. 1991. Adaptive robotic visual tracking. *Proc. of the American Control Conference*. Evanston, IL.:American Autom. Control Council, pp. 962-967.

[36] Papanikolopoulos, N.P., Nelson, B. and Khosla, P.K. 1992. Full 3-d tracking using the controlled active vision paradigm. *Proc. 1992 IEEE Int. Symp. on Intelligent Control (ISIC-92)*. New York:IEEE, pp. 267-274.

[37] Qian, H.P. and De Schutter, J. 1992. Introducing active linear and nonlinear damping to enable stable high gain force control in case of stiff contact. *Proc. 1992 Int. Conf. on Robotics and Automation*. New York:IEEE, pp.1374-1379.

[38] Raibert, M.H. and Craig, J.J. 1981. Hybrid position/force control of manipulators. *ASME J. of Dyn. Sys., Measurement, and Control*. 103(2):126-133.

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

[40] Shirai, S. and Inoue H. 1973. Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*. 5:99-108.

[41] Sharma, R. and Hutchinson, S. 1994. On the Observability of Robot Motion Under Active Camera Control. *Proc. 1994 Int. Conf. on Robotics and Automation*. New York:IEEE, pp. 162-167.

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

[43] Stansfield, S.A. 1988. A Robotic Perceptual System Utilizing Passive Vision and Active Touch. *Int. J. of Robotics Research*. 7(6):138-161.

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

[45] Tarabanis, K., Tsai, R.Y. and Allen, P.K. 1990. Satisfying the Resolution Constraint in the "MVP" Machine Vision Planning System. *Proc. of the 1990 Darpa Image Understanding Workshop*, 850-860.

[46] Tomasi, C. and Kanade, T. 1991. Detection and tracking of point features. Tech. Rept. CMU-CS-91-132. Pittsburgh:Carnegie Mellon University School of Computer Science.

[47] Uchiyama, M., E. Bayo, and E. Palma-Villalon, "A Systematic Design Procedure to Minimize a Performance Index for Robot Force Sensors," *J. Dyn., Sys., Meas., and Control*, 113(9):388-394, Sept. 1991.

[48] Volpe, R.A. and Khosla, P.K. 1991. Experimental verification of a strategy for impact control. *Proc. 1991 IEEE Int. Conf. on Robotics and Automation*. New York:IEEE. pp. 1854-1860.

[49] Weiss. L.E. 1984. Dynamic visual servo control of robots: an adaptive image-based approach. Ph.D Thesis CMU-RI-TR-84-16, Pittsburgh, PA:The Robotics Institute Carnegie Mellon University.

[50] Weiss, L.E., Sanderson, A.C. and Neuman, C.P. 1987. Dynamic sensor-based control of robots with visual feedback. *IEEE J. of Robotics and Automation* RA-3(5):404-417.

[51] Whitney, D.E. 1977. Force feedback control of manipulator fine motions. *ASME J. of Dyn. Sys. Measurement, and Control*. June:91-97.

[52] Whitney, D.E. 1985. Historical perspective and state of the art on robot force control. *Proc. 1985 Int. Conf. on Robotics and Automation*. New York:IEEE, pp. 262-268.

[53] Wilson, W.J. 1993. Visual servo control of robots using kalman filter estimates of robot pose relative to work-pieces. *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, ed. K. Hashimoto. London:World Scientific, pp. 71-104.

[54] Xu, Y., Hollerbach, J.M., and Ma, D. 1994. Force and Contact Transient Control Using Nonlinear PD Control. *Proc. 1994 Int. Conf. on Robotics and Automation*. New York:IEEE, pp.924-930.

[55] Yi, S., R.M. Haralick, and L.G. Shapiro, "Automatic Sensor and Light Positioning for Machine Vision," *Proc. of the 10th Int. Conf. on Pattern Recognition*, 55-59, 1990.

[56] Yoshikawa, T. 1987. Dynamic hybrid position/force control of robot manipulators-description of hand constraints and calculation of joint driving force. *IEEE J. of Robotics and Automation*. 3(5):386-392.

[57] Yoshikawa, T. 1985. Manipulability of Robotic Mechanisms. *Robotics Research 2*, eds. H. Hanafusa and H. Inoue, 439-446, MIT Press, Cambridge, MA.