RELIABLE VISION-GUIDED GRASPING

by

Keith E. Nicewarner and Robert B. Kelley

Rensselaer Polytechnic Institute
Electrical, Computer, and Systems Engineering Department
Troy, New York 12180-3590

August 1992

CIRSSE REPORT #125

Reliable vision-guided grasping

Keith E. Nicewarner and Robert B. Kelley

Center for Intelligent Robotic Systems For Space Exploration
Electrical, Computer, and Systems Engineering Department
Rensselaer Polytechnic Institute, Troy, NY 12180

ABSTRACT

Automated assembly of truss structures in space requires vision-guided servoing for grasping a strut when its position and orientation are uncertain. This paper presents a methodology for efficient and robust vision-guided robot grasping alignment. The vision-guided grasping problem is related to vision-guided "docking" problems. It differs from other hand-in-eye visual servoing problems such as tracking in that the distance from the target is a relevant servo parameter. The methodology described in this paper is a hierarchy of levels in which the vision/robot interface is decreasingly "intelligent," and increasingly fast. Speed is achieved primarily by information reduction. This reduction exploits the use of region-of-interest windows in the image plane and feature motion prediction. These reductions invariably require stringent assumptions about the image. Therefore, at a higher level, these assumptions are verified using slower, more reliable methods. This hierarchy provides for robust error recovery in that when a lower-level routine fails, the next-higher routine will be called and so on. A working system is described which visually aligns a robot to grasp a cylindrical strut. The system uses a single camera mounted on the end effector of a robot and requires only crude calibration parameters. The grasping procedure is fast and reliable, with a multi-level error recovery system.

1 INTRODUCTION

Computer (or machine) vision, and the problems associated with the field, are familiar topics in robotics. While solutions and approaches to static problems such as recognition, perception, calibration, and metrology have flourished, there have been relatively fewer treatments of dynamic issues such as tracking a moving object and visual servoing. Only within the past 5 years has computer technology advanced to the point where the high-speed requirements of these tasks can be met.

There are two basic problems in dynamic machine vision: object tracking and visual servoing. With tracking, we are concerned with locating and tracking one or more moving targets in one or more images. Applications are in air traffic control, military operations, and industrial process control. The camera (or equivalent imaging device) is usually considered stationary and the output is a real-time stream of target locations.

Closely related to tracking is visual servoing, where tracking is used to drive some system parameter to zero. This could mean moving the imaging device to follow a moving target or guiding a robot manipulator to a goal position and orientation. In robotic visual servoing, common tasks include using machine vision as a secondary position sensor (secondary to the robot joint encoders) and visual alignment with an object.

Vision-guided alignment can be applied to such tasks as "docking" with an object and grasping an object. In

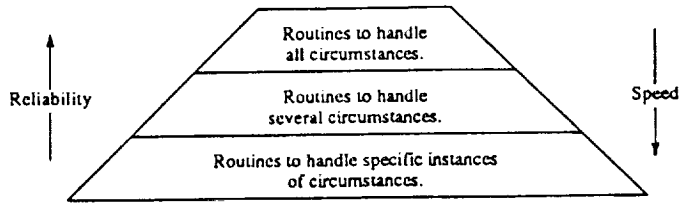

Figure 1: Coordination hierarchy of decreasing reliability and increasing speed.

docking procedures, the robot end effector either is itself a docking mechanism or is holding one. The mechanism is then visually guided to mate with the docking receptacle. In vision-guided grasping, the robot manipulator is visually aligned with an object so that minimal reaction forces and torques result when the gripper is closed.

Thorough image processing invariably requires intensive computation, which in turn requires time. Visual servoing, on the other hand, requires a fast interface between the vision and the robot. We do not generally have the luxury of thorough image processing when it comes to fast, responsive hand-eye coordination. These two needs: rigorous image processing and fast vision updates to the robot are in direct conflict.

To solve this problem, a multi-layered system is presented. This coordination system contains elements of *both* slow, thorough image processing and fast, less rugged image processing. The fundamental concept is that of progressively verifying and taking advantage of more and more assumptions.

The coordination architecture has layers of increasing knowledge at higher levels and decreasing reliability at lower levels. A diagram of the relationships between the layers is shown in Figure 1. This structure allows the necessary assumptions to be verified at higher levels while providing a means for "graceful degradation" from low-level failures.

1.1 Motivations

A proposed construction of the NASA Space Station Freedom involves a large truss structure composed of 2–5 meter struts and reconfigurable nodes. At the Center for Intelligent Robotic Systems for Space Exploration (CIRSSE), we are interested in automating the assembly of these struts and nodes. This problem is studied using a versatile robotic testbed. The CIRSSE testbed consists of:

- 2 9-DOF robots (6 DOF PUMA + 3 DOF linear-track Aronson platform)
- 2 robot grippers equipped with force and cross-fire sensors
- 2 force-torque sensors for each robot wrist
- a pair of cameras mounted on one of the robot grippers
- 2 stationary cameras
- a laser scanner

The stationary cameras and laser scanner can give rough global pose information of the struts in the assembly area. These pose estimates are too rough for such operations as grasping or inserting a strut. The arm cameras provide a means for refining the global pose estimates of struts.

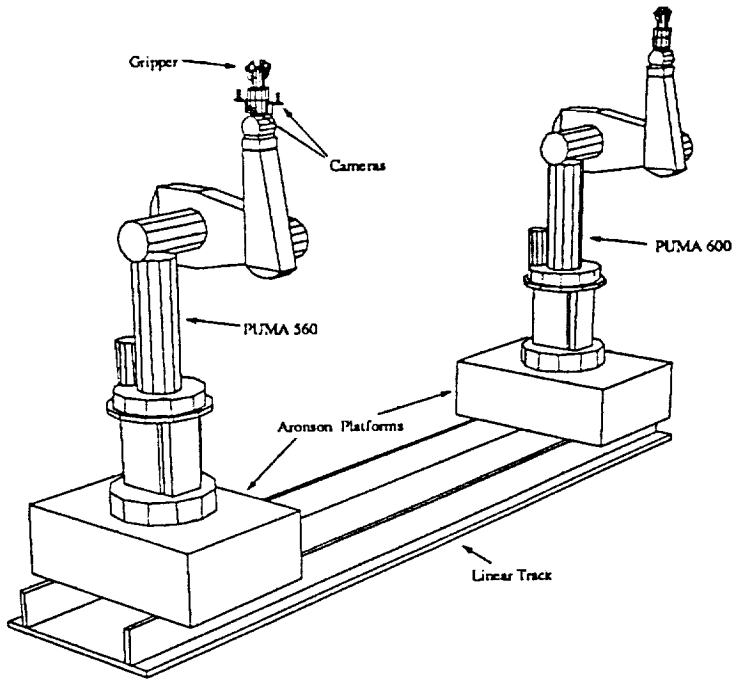

Figure 2: CIRSSE experimental robot testbed.

Figure 2 shows the CIRSSE experimental robot testbed. Note the camera pair mounted on the left robot. Although the vision-guided grasping algorithm discussed in this paper uses only one camera, the two cameras on the arm allow for future research with stereo vision and vision-guided insertion of a strut into a node connector.

2 COORDINATION

Figure 2 shows a flow diagram of the coordination system for strut recognition, visually-servoed alignment, and grasping. Square boxes represent states, rounded boxes represent operations, and arrows represent conditional execution flow. All operations start from the Dead state, where little is assumed about the environment. Two primary flow paths are seen: Grab and Learn. Grab is the "usual" operation of the system, while Learn is a calibration phase which will be described later in this section.

Note that the strut grasping process only works if there is a single strut in the image. If more than one is present, the operator must either select one or adjust the initial pose of the robot such that only one strut is seen. Once a strut has been found, the program must insure that the strut is roughly vertical in the image (within 20° from vertical). This is a requirement for the pose estimation technique discussed by Nicewarner.[1] Once aligned, if the image-plane width of the strut is unexpected, the radius is estimated using the delta-position technique discussed later in this section. If the radius is outside of the range for the specific robot gripper, the strut cannot be grasped and the process fails.

Once we are assured that the camera image contains a valid strut which is roughly vertical, we are ready to visually servo to align for grasping. If a circumferential fiducial stripe is visible, the servoing gains are set such that all 3 translation pose parameters and two of the rotation parameters (rotation about the X-axis and Z axis)

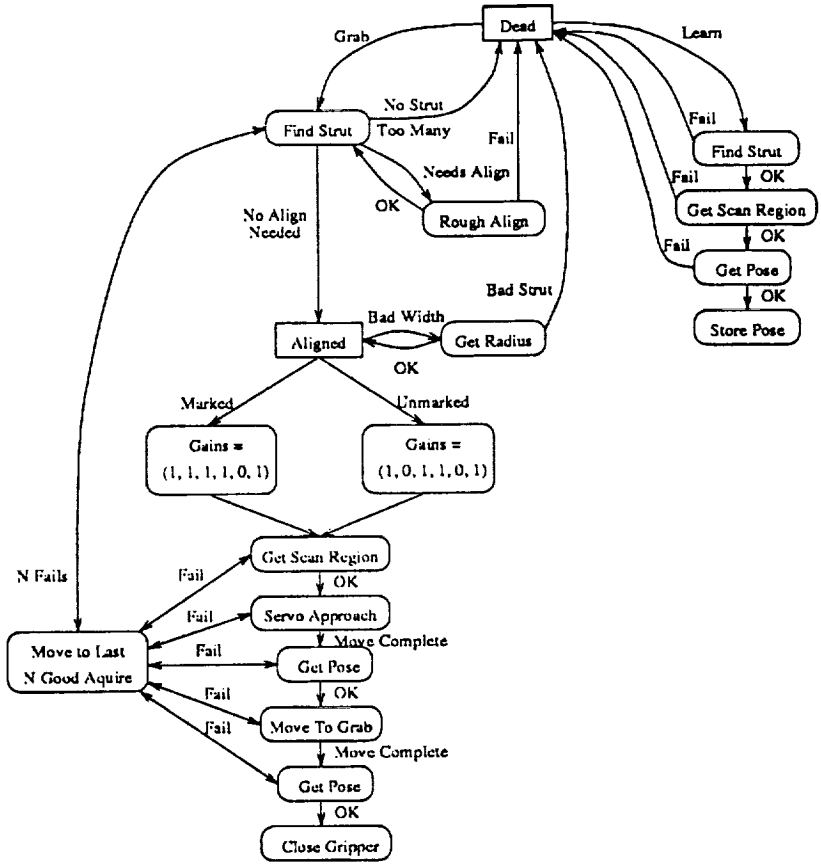

Figure 3: Coordination system for vision-guided grasping of a cylindrical strut.

are used. If a marker is not visible, the Y-axis translation parameter cannot be used, so the servo gains are set appropriately.

Once the servo process begins, if a failure occurs, the robot returns to a previous position where it saw the strut last. If the servo fails there, the robot moves to the next previous position, and so on. After N failures, the program falls back to searching for a strut in the image.

The grasp process ends when the gripper successfully closes on the strut. The operator then must specify what to do with the strut using an external path-planner to place the strut in a desired location or simply move to the robot's "home" position.

The flow-diagram representation of the coordination system is an accurate representation but is more difficult to understand when attempting to convey the basic operation of the system. The coordination system operation can alternatively be thought of as a series of phases. These phases are: learn, recognition, alignment, and approach.

2.1 Learn Phase

Before vision-guided alignment can begin, the target pose for the strut in the camera space needs to be defined. The target pose is defined by simply placing the strut in the gripper and noting the pose calculated by the pose estimator. This procedure is typically done only once as a calibration step whenever the operating conditions of the robot change, such as camera parameters, camera location, lighting, or strut design. Since this is not a time-sensitive task, computation restrictions are not necessary for the image processing.

In a typical learn session, the strut is placed in the gripper and the gripper is closed. An image is then snapped from the camera and the strut is located in the image using the recognition algorithms described by Nicewarner.[1] The pose is then estimated and saved to a file which is from then on loaded and used as the target pose for the strut.

2.2 Recognition Phase

Upon startup, the coordinator assumes nothing about the current image from the camera. First, an image is snapped from the camera and the centroids (or blobs) are extracted. The centroid information not only tells the location of blobs, but also the second moments of each blob. These second moments can be used to obtain a list of blobs which are "long and thin."

Once the long and thin blobs are extracted, collinear blobs are merged together because the fiducial circumferential stripes effectively split a strut into a group of collinear cylinders. The merges are then noted as candidate marker locations, to be later verified.

If no valid struts result from this, the program fails because there are no struts it can see to be grasped. If there is more than one strut in the image, the program fails as well because there is no criterion to choose an appropriate strut to grasp. The program only continues if there is one valid strut in the image.

The information so far can be used to crudely center and align the strut vertically in the image. As stated before, vertical alignment is necessary for the pose estimation algorithm. This rough alignment is done simply by calculating the delta movement in the image plane for the marker and strut axis using the information given by the strut recognition routine.

The next verification made is that the radius of the strut is within an expected range. The radius of the strut can be estimated by observing the change in the image induced by moving the robot a certain distance towards the strut. If the radius projected onto the screen at the first position is r_1 and the projected radius at the second position is r_2, the radius R can be determined by similar triangles.

$$\frac{R}{d_1} = \frac{r_1}{f} \tag{1}$$

$$\frac{R}{d_2} = \frac{r_2}{f} \tag{2}$$

where f is the focal length of the camera and d_1, d_2 are the distances from the strut to the camera focal point at the two positions. Recognizing that $d_2 = d_1 + \Delta d$, we can solve these equations for R,

$$R = \frac{\Delta d}{f}\left(\frac{r_2 r_1}{r_2 - r_1}\right) \tag{3}$$

Therefore, we can use the calibration of the robot to move a given distance and calculate an estimate of the strut's radius. If this strut is outside of an expected range, the program fails because the object most likely is a bogus object.

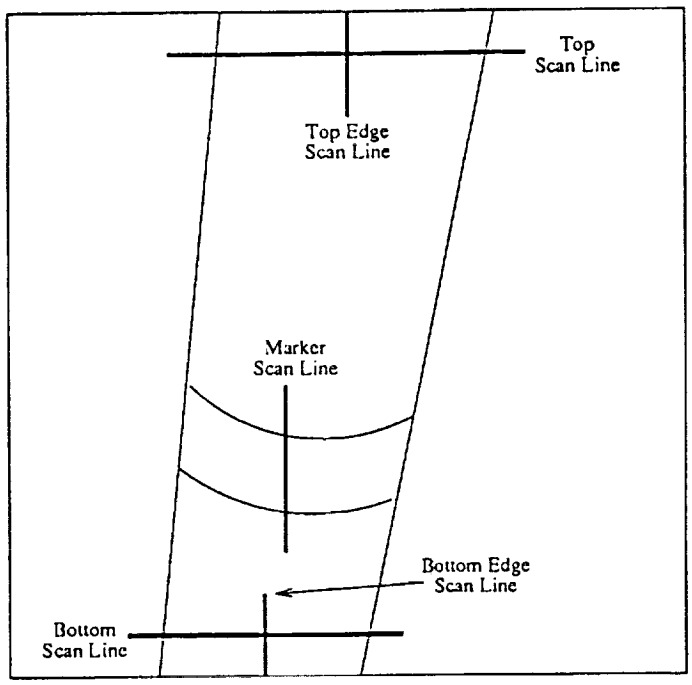

Figure 4: Scan lines composing a typical scan region.

Once the strut has been verified, the critical processing areas of the image are chosen. These critical areas are called *scan lines* and are processed with a 1-D edge detector for rapid pose estimation. Five scan lines comprise a *scan region* (see Figure 4). These scan lines are either vertically or horizontally oriented to provide fast access to the critical areas of an image by a computer. Two scan horizontally along the top and bottom of the screen for edges. Two more scan vertically across the top and bottom edges to detect the end of the strut. The last scan line vertically crosses the fiducial marker (if present).

The scan line positions and scan ranges are chosen to minimize the noise that might be encountered during the alignment phase. The top and bottom horizontal scan lines are chosen to be as far apart as possible to ensure more accurate pose estimations. The top and bottom cross scans are used to ensure that the top and bottom horizontal scans are sufficiently far from the end of a strut (if visible).

2.3 Alignment Phase

The alignment phase begins by rapidly processing the scan lines for edges, or *critical points*. There are five critical points: 2 on the top horizontal scan line, 2 on the bottom horizontal scan line, and 1 representing the mid-point of the edges across the fiducial stripe. If some unexpected noise is encountered while scanning for critical points, the scan ranges and scan line positions can be adjusted. It can be shown[1] that 5 of the 6 strut pose parameters can be determined from only 5 critical points. The pose of the strut is computed relative to the camera. The 5

pose parameters are:

1. R_x – the tilt angle of the strut axis out of a plane perpendicular to the optical axis.
2. R_z – the clockwise rotation of the strut about the optical axis, relative to the image plane y-axis.
3. T_x – the horizontal displacement of either the strut marker or the center of the strut from a vertical plane through the optical axis.
4. T_y – the vertical displacement of the strut marker (if visible) from a horizontal plane through the optical axis.
5. T_z – the distance from the camera lens to the center of the strut along the optical axis.

Note that R_y is not available since the strut is rotationally symmetric. T_y is only available if a stripe is visible; otherwise, only four parameters are used. Effectively, if no stripe is seen, the strut will be grasped arbitrarily along the axis.

For the alignment phase, the robot controller servoes all the parameters except T_z to zero. The distance is servoed to an optimal distance from the strut. This distance is determined primarily by the focal depth and field of view of the camera.

2.4 Approach Phase

Ideally, the alignment phase could be continued all the way to the target pose. Because the image detail increases as we get closer, the pose estimates become more accurate, so we should expect our best performance when the strut is grasped. In actuality, although the pose estimate errors do indeed decrease as the distance decreases, the sensitivity of the critical point extraction process increases. As the strut projection becomes larger in the image, unavoidable minute "jerks" in the robot's movements can cause the feature extraction process to fail.

To solve this problem, the visual servo process halts when the last pose estimate is the "best." From there, the robot moves "blindly" to grasp the strut. Weighing the relative costs of completely servoing versus the loss in fault tolerance introduced by blind motion is discussed by Nicewarner.[1]

3 IMPLEMENTATION

The vision-guided grasping systems discussed in this paper was successfully implemented with the CIRSSE experimental testbed shown in Figure 2. The layout for CIRSSE computing resources used in this paper is shown in Figure 5. There are three primary platforms: the UNIX host computer, the vision VME cage, and the motion control VME cage. The platforms are interconnected via an ethernet network.

A Sun 4 computer is used as the UNIX host and executes the high-level coordination software. The vision VME cage contains:

- 1 Motorola MV-147 processor
- 1 Motorola MV-135 processor
- 8 special-purpose Datacube DSP boards

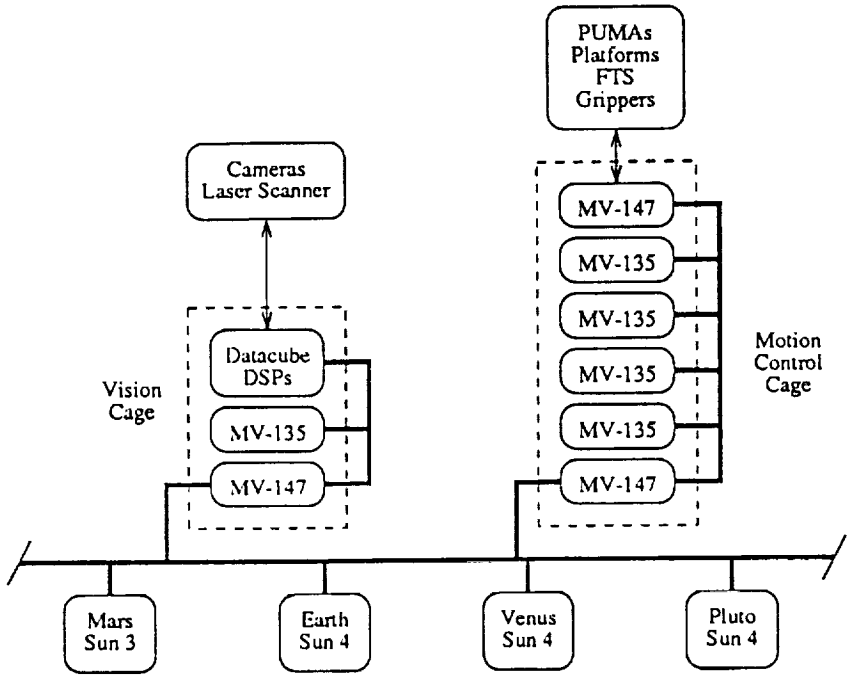

Figure 5: CIRSSE testbed resource layout.

- interface to laser scanner

The motion control VME cage contains:

- 2 Motorola MV-147 processors
- 4 Motorola MV-135 processors
- interfaces to grippers and force-torque sensors
- interfaces to Unimation controllers

Both VME cages are running under the Wind Rivers VxWorks real-time operating system.

The necessary high-speed communications between the vision and motion control cages was implemented using BSD UNIX datagram sockets as opposed to using standard stream sockets. Stream sockets buffer data packets and insure reliable transmission. Datagram sockets have no such features and as a result are much faster yet less reliable. In our implementation, data packets are lost on occasion, in which cases the trajectory generator assumes the pose of the strut relative to the camera has not changed. This could potentially lead to untimely jerks in the robot motion when the transmissions are restored. However, since consecutive pose estimates are relatively close together, no adverse effects are observed.

4 RESULTS

Several experiments were performed to evaluate the performance of the vision-guided grasping system presented in this paper. In one experiment, white cylinders of various diameters (8 mm, 16 mm, 22 mm, and 38 mm) were used to test the pose estimation process with respect to the robot calibration. The results of this experiment are discussed by Nicewarner.[1]

Another set of experiments performed involved finding and grasping a strut, moving to "home" position, then placing the strut randomly and repeating the process. This is perhaps the best measure of performance for our system because it conveys the reliability and *repeatability* of the process. With the completed system, around 100 trials were made. All were handled properly, meaning that if the strut was not visible in the starting image, the program exited and if the strut *was* visible, is was successfully grasped.

5 CONCLUSIONS

A multi-layered vision-guided grasping system has been presented which successfully resolves the conflict between rigorous image processing needs and rapid vision updates to the robot. This system contains elements of *both* slow, thorough image processing and fast, less rugged image processing. The fundamental concept is that of progressively verifying and taking advantage of more and more assumptions. The coordination architecture has layers of increasing knowledge at higher levels and decreasing reliability at lower levels. This structure allows the necessary assumptions to be verified at higher levels while providing a means for "graceful degradation" from low-level failures.

A two-level vision system for vision-guided grasping has been discussed which handles both high-level strut recognition and low-level rapid strut pose estimation. The recognition is performed based on the moments of inertia of the strut segment projections. The rapid pose estimation method described is unique for cylindrical objects. It exploits the fact that only 4 edges on parallel scan lines are needed to estimate 4 of the pose parameters. With the addition of a simple fiducial stripe around the strut, we can estimate the 5 pose parameters necessary for grasping the strut in a particular location along its axis. The pose estimation runs easily at frame-rate and is reasonably accurate under a wide range of operating conditions. The method is relatively insensitive to camera model uncertainties and can be easily calibrated in a one-step procedure.

The overall design is modular so that lower modules can be changed without significantly effecting the operation. This means that the vision-guided grasping system can be ported to a different robot system and operate in a different environment. In addition, the multi-layered architecture provides robustness and fault-tolerance—qualities that are demanded of space-worthy systems.

6 ACKNOWLEDGMENT

This research is performed at the Center for Intelligent Robotic Systems for Space Exploration at Rensselaer Polytechnic Institute and is supported in part by Grant number NAGW-1333 from NASA.

7 REFERENCES

[1] K.E. Nicewarner. *Competent Vision-Guided Grasping.* Masters Thesis, Rensselaer Polytechnic Institute, Elec., Comp., and Sys. Eng. Dept., Troy, NY 12180-3590, May 1992.

[2] Robert J. Schalkoff. *Digital Image Processing and Computer Vision*. John Wiley & Sons, Inc., New York, NY, 1989.

[3] B.K.P. Horn. *Robot Vision*. The MIT Press, Cambridge, MA, 1989.

[4] R.Y. Tsai. A versatile camera calibration technique for high accuracy 3-d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE J. Robotics and Automat.*, RA-3:323–344, 1987.

[5] J.R. Noseworthy. *Inaccuracies in three-dimensional vision systems — theory and practice*. Masters Thesis, Rensselaer Polytechnic Institute, Elec., Comp., and Sys. Eng. Dept., Troy, NY 12180–3590, August 1991.

[6] J.T. Fedemma and O.R. Mitchell. Vision-guided servoing with feature-based trajectory generation. *IEEE Transactions on Robotics and Automation*, 5(5):691–700, 1989.

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

[8] D. Vernon and M. Tistarelli. Using camera motion to estimate range for robotic parts manipulation. *IEEE Transactions on Robotics and Automation*, 6(5):509–521, 1990.

[9] C. Lee and W. Lin. A hybrid method of visual guiding for eye-in-hand robot. In *European Control Conference*, pages 2524–2529, 1991.

[10] M.J. Korsten and Z. Houkes. Parametric descriptions and estimation, a synergetic approach to resolving shape from shading and motion. In *Third International Conference on Image Processing and its Applications*, pages 5–9. IEE, 1989.

[11] M. Richetin, M. Dhome, and J.T. Lapreste. Inverse perspective transform from zero-curvature curve points application to the localization of some generalized cylinders. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 517–522. IEEE, 1989.

[12] H. Printz. Finding the orientation of a cone or cylinder. In *Proceedings of the IEEE Computer Society Workshop on Computer Vision*, pages 94–99. IEEE, 1987.

[13] Y. Shirai and H. Inoue. Guiding a robot by visual feedback in assembly tasks. *Pattern Recognition*, 5:99–108, 1973.

[14] N.P. Papanikolopoulos, B. Nelson, and P.K. Khosla. Monocular 3-d visual tracking of a moving target by an eye-in-hand robotic system. *Technical Report, Carnegie Mellon University, The Robotics Institute*, 1991.

[15] K.E. Nicewarner and R.B. Kelley. Efficient visual grasping alignment for cylinders. In *Proceedings of the SPIE Conference on Cooperative Intelligent Robotics in Space II*, volume 1612, pages 161–171, Boston, MA, November 1991.

[16] K.E. Nicewarner and R.B. Kelley. Vision-guided gripping of a cylinder. *Reprints First IARP Workshop on Robotics*, June 1991.