Distributed Visual Servoing with a Roving Eye

David Hershberger
Robotics Institute
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
hersh@ri.cmu.edu

Robert Burridge
S&K Electronics

David Kortenkamp
Metrica, Inc.

TRACLabs
1012 Hercules Drive
Houston, TX 77058

Reid Simmons
Robotics Institute
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213

Abstract

This paper presents experimental results of preliminary research into multi-robot coordination for construction tasks. Experiments demonstrate that an autonomous "roving eye" robot can provide feedback to a manipulator to align targets from a wider variety of situations than is possible with fixed cameras, without sacrificing the accuracy provided by cameras at close range. The roving eye changes its location autonomously based on current images of the manipulated object and target, always striving for the best view of the task.

1 Introduction

An important type of cooperation used frequently in construction projects can be viewed as a remote sensing operation. When a heavy steel beam needs to be attached to the structure of a new building, a crane operator can move it to roughly the correct position, but does not have the visual acuity (from a long distance) or the dexterity (through the crane) to do the final placement of the beam. Workers near the beam provide high acuity visual feedback to the crane operator and then grab the beam to pull it into the final position.

We are investigating architectures for robot coordination for use in autonomous assembly of large structures. A crane robot will provide heavy-lift capability while a smaller more dextrous robot manipulator provides fine motion control for assembly, and a third robot acts as an observer, providing visual feedback to the crane and fine manipulator. NASA is funding this research to provide the technology for a multi-robot construction team for the assembly and maintenance of a Mars base. Terrestrial applications include construction in hazardous environments and eventually construction in ordinary building projects.

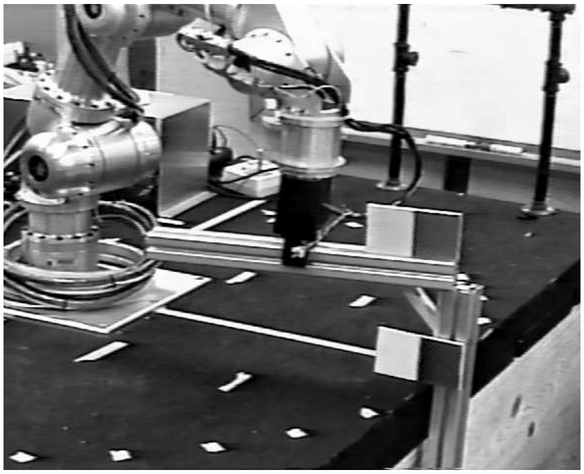


Figure 1: The manipulator visually aligning a beam with a fixed structure. Visual feedback comes from tracking the rectangular fiducials.

As a first step, we have implemented a distributed visual servoing system consisting of a robot arm and a "roving eye" robot. The roving eye is a mobile robot with a pair of cameras that provides visual feedback to the arm controller by tracking bi-colored rectangular fiducial marks. Mobile cameras can track the fiducials in a wider variety of situations than is possible with fixed cameras, without sacrificing the resolution provided by cameras at close range. In the initial phase of this project the assembly task has been simplified to that of aligning a beam with some existing structure. Figure 1 shows the robot arm aligning the end of one beam with another. The desired alignment is specified by a relative pose between the two fiducials.

Our approach consists of two main servo loops: the visual servoing of the arm, and the motion of the roving eye. The visual servoing is an explicitly distributed task in which one robot sees and the other acts. This raises issues of synchronization and reference frames because each robot must stay aligned with the other in time, plus the position information they exchange must be meaningful to both of them. The motion of the roving eye robot serves to keep a good view of the fiducials. It is coupled to the robot arm indirectly, through the position and motion of the fiducials.

For simplicity in this initial implementation, synchronization in the visual servoing of the arm is handled with a look-then-move scheme. For each iteration, the vision system on the roving eye looks at the scene and calculates the error between where the moving object is and where it should be. It scales this down by a gain factor and sends it to the manipulator. The manipulator moves the object the amount indicated, stops, then sends a message to the roving eye indicating it has finished the move. The cycle repeats until the alignment is complete. This approach reduces problems caused by failures or delays in the communication system that could allow the manipulator to damage itself or the task objects. It also minimizes minimizes the effects of arm dynamics in the control problem.

The reference frame used to communicate position feedback from the roving eye to the manipulator is attached to observations of the object being manipulated. This way the position measurements are always relative to the manipulator, eliminating the need to calibrate the position of the roving eye.

The roving eye is controlled with three behaviors: panning to center the fiducials in the images, zooming to move the cameras as close as possible to the fiducials, and lateral motion to face the fiducials as directly as possible. Together these behaviors keep the roving eye directly in front of the fiducials and as close as possible without losing them from the field of view.

The remainder of this paper describes the distributed visual servoing in more detail. Section 2 describes related work in the fields of vision and visual servoing control. The design and implementation of the system are presented in Section 3. Section 4 presents experimental results, and future work is discussed in section 5.

2 Related work

The literature on visual servoing is extensive, and is only briefly discussed here. Hutchinson, Hager, and Corke [4] provide an excellent tutorial on visual servoing along with an extensive bibliography. In particular, they describe a taxonomy of visual servo control architectures with three important distinctions. The first is whether a vision system provides Cartesian set points for a robot's joint-level controller (called "dynamic look-and-move"), or whether it directly computes the joint inputs. Dynamic look-and-move is especially appropriate for our distributed visual servoing because of possible unpredictable network lag in the inter-robot communication. Systems of the second type typically require very high speed vision to ensure stability and present difficult coupled dynamics.

The second distinction is whether the error signal is defined in 3D task space coordinates (*position-based* control) , or directly in terms of image features (*image-based* control). While image-based control is attractive, we chose position-based control because of the difficulty of specifying target part alignments at run time.

The third distinction is between *endpoint closed-loop* (ECL) systems that observe both the target object and the robot end-effector and *endpoint open-loop* (EOL) ones that only observe the target object. EOL may often be simpler to implement because only one object must be tracked visually. However for the construction task, ECL seemed more appropriate and simpler, because the grasp of the beam in the gripper may not be known precisely, preventing an EOL system from properly aligning the beam.

XVision [3] is a C++ class library for computer vision. It provides two types of objects that were used directly in the current work: tracking primitives and a container class that joins multiple objects. The basic tracking primitives are intensity edge trackers, blob trackers, and image patch trackers. The container class is used to build the corner tracker class (provided with XVision), which combines a pair of edge trackers at an angle to each other. The resulting class can then define constraints between its members, such as the corner class that constrains one end of each of the two edges to be at the same point. For visual tracking of the rectangular targets in the current work, four corner trackers were combined into a rectangle tracking class.

Nelson and Khosla [5] present a method for calculating "observability ellipsoids" that represent the resolving power of geometric configurations of cameras. Ellipsoids are six dimensional, giving resolving power in each of the 6 degrees of freedom of rigid body motion in three dimensional space. They describe how to calculate the ellipsoids and present a method for choosing camera positions to get the best overall resolving power. For our application at this point, we have available only the two stereo cameras on our roving eye robot. The observability ellipsoids consistently show that stereo with a wider baseline relative to the target produces better resolution in all dimensions, up to having the cameras looking at the scene from 90° apart. This corresponds to our results that the tracking performance in the camera depth dimension is better when the roving eye is close to the fiducials, providing a greater angle between the camera views.

Wang and Wilson [6] describe a system using a Kalman filter to track the motion of an object in 3D. Their system tracks the 3D position, orientation, and motion of an object seen by a single camera in a sequence of images. For best performance it requires at least 5 observable non-coplanar feature points on the tracked object. They mention that the nature of the implemented Kalman filter requires that values in the Q matrix be adjusted depending on the speed of the tracked object, as accelerations of the object were modelled as noise.

3 Implementation

This section describes details of the fiducial-tracking visual servoing system and control of the roving eye. The tracking system was designed to use relative measurements where possible, to reduce the need for accurate calibration. This is especially useful because of the roving eye: absolute positions of the objects would require keeping accurate track of the roving eye's pose. The main measurement calculated is the 6 DOF pose of the fixed fiducial with respect to the moving fiducial. With an appropriate gain setting, this method can converge to the correct value even when the relative pose measurements have significant calibration errors, because each motion merely needs to move the object in the right general direction.

3.1 Visual Servoing

The five components of the vision processing are: color filtering, blob finding, corner tracking, calculating relative depths, and model fitting. After model fitting, the relative pose between the fiducials is sent to the arm controller, and information for moving the roving eye is sent to the roving eye. Figure 2 shows the relationships between these components. Communication between components on a robot is facilitated by the skill manager component of the 3T architecture developed at NASA JSC [2]. The communication between the robots is sent over radio ethernet using the IPC package developed at Carnegie Mellon for sending structured data.

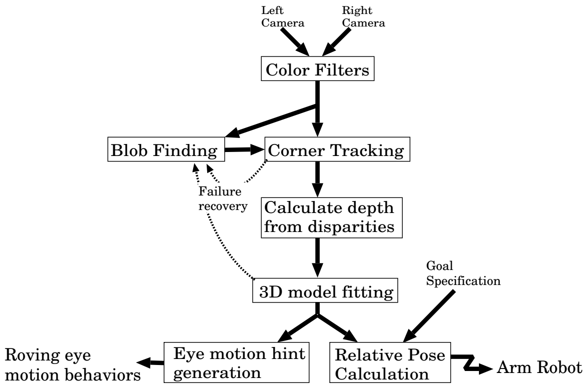


Figure 2: Design of the fiducial tracking system.

The fiducials used for tracking are planar rectangles. Figure 1 shows the pattern of coloration: two adjacent solid colored squares. There are 4 color filters, one for each color of each fiducial. Each filter looks for pixels with hue between high and low limits and saturation and intensity above preset thresholds. Initializing the corner trackers for a fiducial begins by finding the largest blob of each color. If the two blobs are adjacent and square, the corner trackers' locations are set to the outside corners. This routine is relatively slow even on subsampled images, so it is used only for initialization and to recover from corner tracking failures. Once initialized, the corner tracking primitives from XVision track the outer 4 corners of the two-colored rectangles. When the corner trackers are getting good results and the resulting 3D points match the fiducial models well, the vision processing is quite fast because only the local areas around the corners need to be processed.

Once the positions of these corners are found in the left and right images, they are used to calculate disparities. In order to remove the requirement for precisely calibrated camera vergence, relative disparities are used, rather than absolute disparities. This gives the relative depths between the 3D points, rather than the absolute depth from the camera. Without absolute depth, information about the size of the objects is lost. To compensate, the scale of the observed corner points are modified to match the scale of the corner points of the models.

The model used for relative disparity is that an arbitrary offset has been added to every observed disparity in an image. This is approximately what happens when the angle between a stereo pair of cameras is changed. To enable depth calculations, the minimum disparity is set to a fixed value, and all the other disparities are adjusted by the same amount. The result is that the angles of the cameras can be moved with no effect on servoing performance. However even with the scale factor correction the computed relative depth values are not strictly correct: they are an approximation valid for fairly small relative disparities and for a given fixed camera separation. When the stereo baseline is changed, a parameter in the distance-from-disparity calculation needs adjusting.

Once the relative depths of the points are found, a model fitting technique by Arun, Huang, and Blostein [1] based on SVD is used to find the poses of the models of the two fiducials that fit best with the 3D positions of the tracked corner points. This model fitting technique is simple and fast because the correspondences between the model points and the observed points are already known.

Once the fiducial models have been aligned with the observed corner positions, the 6D relative pose of the fixed fiducial relative to the moving fiducial is calculated, time-filtered (to reduce sensing noise), multiplied by a small gain (usually 0.25), and sent to the arm controller (the new version of the software includes the gain in the arm controller rather than the vision code). The arm controller then calculates a new gripper pose by composing the commanded pose offset with the current gripper pose. When the motion is complete, it signals the roving eye to request another offset.

3.2 Roving Eye Control

Control of the roving eye is accomplished with three primary behaviors: a panning behavior to keep the fiducials centered in the images, a zooming behavior to move the cameras as close as possible to the fiducials, and a lateral motion behavior to move to face the fiducials as directly as possible. Running together, these behaviors keep the roving eye directly in front of the fiducials and close enough to see them well, but not so close that they are in danger of moving outside the field of view of the cameras. The behaviors are diagrammed in figure 3a, and the resulting motion is depicted in figure 3b.

The roving eye behaviors receive information from the vision system in the form of "eye motion hints". These consist of the bounding box of the fiducials in the images and the average angle of the surface normals relative to the camera pointing angle. The bounding box of the fiducials is used by the panning behavior to keep the edges of the fiducials as far as possible from the edges of both fields of view simultaneously. This bounding box is also used by the zooming behavior that drives the roving eye towards or away from the fiducials. If any side of the bounding box is too close to the edge of the frame, the roving eye backs away. If all sides are too far from the edges of the frame, it drives forward. There is a dead band in between to prevent noisy measurements from causing oscillations.

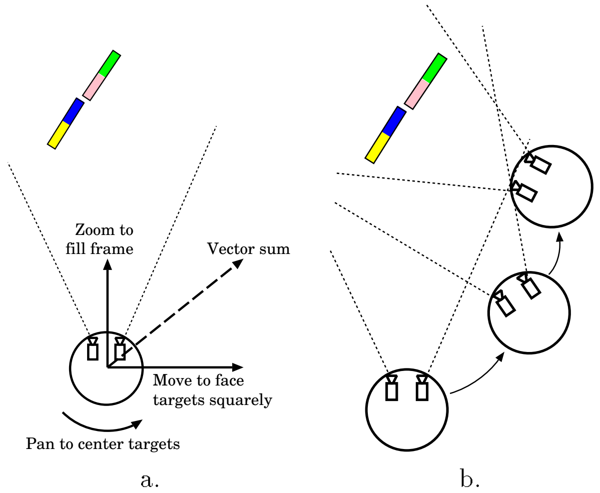


Figure 3: a. The three motion behaviors of the roving eye robot. b. The resulting motion of the roving eye.

The lateral motion behavior uses the average of the two fiducial surface normal angles projected onto the horizontal plane. It tries to move the robot to the left or right relative to the view direction of the cameras in order to be most directly in front of the fiducials. This is important since the fiducials are planar and one-sided. When viewed from an angle that is too steep, the corner tracking and blob-finding algorithms fail.

The three roving eye behaviors combine to produce smooth motion when the vision updates are fast enough relative to the driving speed of the roving eye. Figure 3a shows how the lateral motion and zooming behaviors' outputs are combined in a vector sum. These vectors are defined relative to the orientation of the cameras so that when the panning behavior turns the cameras, the directions of the vectors from the other behaviors change accordingly. Lateral robot motion moves the fiducials off-center in the images, triggering the panning behavior. Together these two effects generate smooth motion in a spiral arc.

4 Experimental Results

This section presents the setup and results of three experiments characterizing the performance of the system. The first experiment demonstrates the superior servoing accuracy available from the system when used with moving cameras compared to the same system with fixed cameras. The second experiment shows the repeatability of the motion of the roving eye robot. The third tests the upper limit of visual servoing and roving eye speeds. For all these experiments, the servoing of the arm was run in a 4 DOF space of control but the target fiducial was only ever situated vertically on the surface of the manipulator's table. The experimental data therefore only measures x, y and yaw. Separate experiments highlighting the manipulator's z axis motion are not critical: the z axis performance is similar to the y axis performance because both are largely parallel to the cameras' image planes.

The experiments were performed at NASA JSC using a Nomad 200 for the roving eye with two Sony 990 cameras mounted on top approximately 20 cm apart. The manipulator is a 5-axis roll-pitch-pitch-pitch-roll arm developed at Metrica, Inc. and delivered to NASA under SBIR number NAS9-97009 with an Eshed parallel jaw gripper. The actual motion of the arm was limited to 4 DOF in which it has dextrous control: x, y, z, and yaw (rotation about z, which is vertical). Independent instances of 3T [2] were used as the software architecture for all three agents (vision, roving eye platform, manipulator), and run on three pentium PCs running Linux.

Figure 4 shows the layout of the servoing accuracy experiment. All the target fiducial poses are located within a small area because of the need to allow the system to work with fixed cameras for comparison with the roving eye version. The target fiducial was moved from pose 1 through pose 10 sequentially, with the actual servoed positions recorded from the arm's joint angles. 10 seconds were given for the arm to come to rest at each location, which was more than enough in most cases. The set of 10 poses were run 5 times each for the fixed eye case and the roving eye case. The tracking software for the fixed eye case was exactly the same as that for the roving eye, but with the roving eye motion turned off.

The fixed eye system was never able to see target pose 6 (the most steeply angled pose), and missed others occasionally as well, leaving a total of 43 successful servoing trials. The roving eye system missed pose 6 once, for a total of 49 successful trials. The accuracy results are summarized in table 1. The smaller standard deviations of the roving eye version demonstrate the increased precision available when the cameras are closer to the fiducials. The Wilcoxon rank sum test on the absolute values of the x and y dimension positioning errors gives a 98.5% and 93.9% probability respectively that the errors came from different distributions, demonstrating that these results are statistically significant. The same test gives only a 6.5% probability for yaw. This small differentiation may be due to a systematic error affecting both systems: a small drift in the yaw control of the manipulator was discovered later.

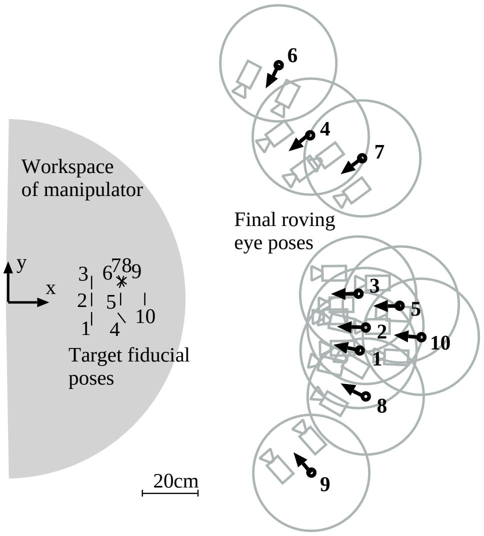


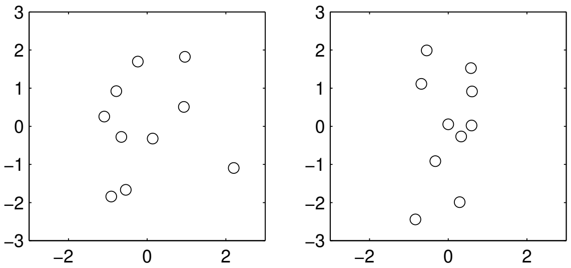
Figure 4: Layout of experiment. The right half of the manipulator's workspace is shown in gray on the left. The target fiducial poses are shown within this, and the final poses of the roving eye robot for each fiducial pose are on the right.

The closest approach that the roving eye came to the fiducials was roughly half the distance at which the cameras were positioned for the fixed eye trials. This closer approach gives better video resolution of the fiducials, but since the cameras can move, it does not sacrifice the size of the workspace like moving the fixed cameras closer would. The increased error bias in x for the roving eye is likely due to inaccuracies in the relative depth calculation that show up at close range (a more recent version of the system uses absolute depth, which eliminates this problem).

The second experiment demonstrates that the motion behaviors of the roving eye generate repeatable motions when given the same fiducial poses. The target fiducial was cycled 10 times between poses 4 and 5 (from figure 4), and each time the roving eye and manipulator were allowed 10 seconds to come to rest. The results presented in figure 5 show the roving eye returning to within roughly 2cm of its original posi-

cameras	$\sigma_{\mathbf{x}}$	$\sigma_{\mathbf{y}}$	$\sigma_{\mathbf{yaw}}$
Fixed	9.6mm	6.5mm	4.5°
Roving	5.5mm	4.4mm	2.7°
	mean x error	**mean y error**	**mean yaw error**
Fixed	0.3mm	3.6mm	−3.6°
Roving	1.1mm	3.2mm	−3.7°

Table 1: Servoing accuracy comparison. The σ values are the standard deviations of the pose errors. The mean errors show the bias of the errors for each dimension.



pose 4.

pose 5.

Figure 5: Plots showing the distribution of final locations of the roving eye for target fiducial poses 4 and 5. Units are centimeters from the centroid.

tion each time after moving to a location roughly 60cm away.

To determine roughly how fast the system can respond to fiducial motion, a third experiment was run in which the target fiducial was moved in an arc around the workspace of the manipulator at different speeds. The arc had a 50cm radius centered around the base of the manipulator, with the target fiducial moved in increments of 10° (about 8cm along the arc). With 10 seconds between each move, the manipulator and the roving eye were both able to keep up through 150° of arc, limited only by the available space around the manipulator table in the lab. With 7 seconds between each move, the visual servoing of the manipulator was unable to keep up, thus separating the fiducials farther and farther. The separating fiducials drove the roving eye to back away farther and farther to keep them in its field of view, eventually losing sight of them.

Achieving high servoing frequency was not a primary goal of this research, especially with a wireless network as part of the main servo loop. Without careful optimization of the code, the vision system ran at about 4 Hz. When the time for the arm to complete one commanded motion is included, the total servo loop speed is about 0.5 Hz. This is slow for a servo system, but because the arm currently stops in between each commanded motion (as compared to a velocity control scheme), it is safe from overshooting due to lost messages. Given a higher performance vision system and a visual servo loop that yields continuous motion, the system would likely be able to keep up with much faster target motion.

5 Future work

Continuing work is under way to expand the system to control both a robotic crane and a mobile manipulator, such that the crane provides heavy lift capability, the mobile arm provides precise motion control, and the roving eye provides visual feedback to both. Figure 6 shows a simulation from this ongoing work. Another extension will add more cameras, and incorporate the additional information to improve the servoing accuracy and increase the visible workspace volume. It will sometimes be the case that fiducials will be far enough apart that they cannot be tracked in the same camera frame. An attention mechanism may solve this by taking turns tracking one and then the other until they approach each other. An important practical concern is to integrate the roving eye behaviors with an obstacle avoidance behavior. There was some difficulty keeping the existing behavior from disrupting the visual tracking, which requires small careful movements near obstacles.

Another important area of future work on this project is that of coordination of the several real robots planned, plus a larger number of simulated robots. Issues such as how teams are created and disbanded and how jobs are allocated to teams and individuals will be addressed once more robots are involved, both in simulation and with the real robots.

6 Conclusions

This paper presents work toward enabling multi-robot construction projects for NASA Mars missions and terrestrial applications. The implemented system uses relative poses in task space to make performance independent of relative robot positions, requiring little sensor calibration. The roving eye motion behaviors allow operation in a wider variety of situations than is possible with fixed cameras, without sacrificing the high resolution provided by cameras at close range. The same basic system has been implemented on three

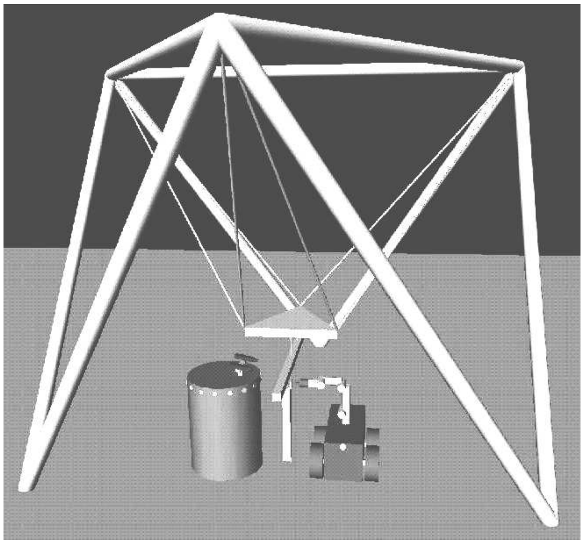


Figure 6: A simulation of the robocrane, mobile manipulator, and roving eye.

different robot platforms now, including a pair of mobile robots, the mobile robot and fixed arm system described here, and a simulation of a robotic crane and mobile robot. Continuing work will expand the system to groups of three or more robots cooperating to assemble large structures.

Acknowledgments

The authors would like to thank Robert Ambrose, the designer of the manipulator arm, Robert Hirsh for his work on the color filtering, the Robotics and Automation Lab at NASA Johnson for hosting the research, and Greg Armstrong for his help with the hardware of an early version of this work. This research was sponsored by NASA contract NAG2-1294.

References

[1] K. Arun, T. Huang, and S. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[2] R. Bonasso, R. Firby, E. Gat, D. Kortenkamp, D. Miller, and M. Slack. Experiences with an architecture for intelligent, reactive agents. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2–3):237–56, 1997.

[3] G. Hager. Xvision: Combining image warping and geometric constraints for fast visual tracking. *Computer Vision — ECCV '96, Spring Verlag Lecture Notes in Computer Science*, (1064):507–517, 1996.

[4] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–70, 1996.

[5] B. J. Nelson and P. K. Khosla. The resolvability ellipsoid for visual servoing. In *Proceedings of the 1994 Conference on Computer Vision and Pattern Recognition (CVPR94)*, 1994.

[6] J. Wang and W. J. Wilson. 3d relative position and orientation estimation using kalman filter for robot control. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 2638–2645, 1992.