ACTIVE, UNCALIBRATED VISUAL SERVOING

Billibon H. Yoshimi and Peter K. Allen *
Center for Research in Intelligent Systems
Columbia University New York, NY 10027

Abstract

We propose a method for visual control of a robotic system which does not require the formulation of an explicit calibration between image space and the world coordinate system. Calibration is known to be a difficult and error prone process. By extracting control information directly from the image, we free our technique from the errors normally associated with a fixed calibration. We demonstrate this by performing a peg-in-hole alignment using an uncalibrated camera to control the positioning of the peg. The algorithm utilizes feedback from a simple geometric effect, rotational invariance, to control the positioning servo loop. The method uses an approximation to the Image Jacobian to provide smooth, near-continuous control.

1 INTRODUCTION

In many real world applications, there exists a need to align one object with another. Many researchers have concentrated their efforts on recovering the true position (the world coordinates) of the object to be manipulated. They used these results to generate movement commands for their servoing systems. The results of their research were a body of techniques which required a precise calibration between the camera and world systems to insure adequate performance. Typically, these calibrations are only accurate in a small subspace of the robot's workspace. As the robot leaves the region where the calibration occurred, accuracy degenerates quickly.

Instead of recovering the calibration between the real world and the image plane, our algorithm exploits a transformation which converts a 3-D quantity (positional location) into a 2-D effect (positional invariance under rotation). The traditional mapping problem (also known as the calibration problem) determines the position of objects based on relative scale difference, perspective distortion, and/or several other relationships which exist between a calibrated system and an observing system. Several methods exist for finding this calibration, but none exploit the fact that a known movement of the camera system can result in useful motion information in the image system without knowing the exact calibration between the systems.

We approach the problem of visual servoing through a classic, robot-control problem: how to insert a peg into a hole using only vision to control the peg's movement. We propose a new technique which will allow the robot system to maintain an arbitrary, geometric relationship with an object system, and as a result of certain operations, the robot-object system can "calibrate itself to" or "can define its location with respect to" the unknown camera system. Our system "calibrates" itself while performing the useful task of moving to the goal position without ever really knowing the true location of the camera system. In this respect, it is possible to servo to a location based purely on the object's change in image coordinates. Given a camera which rotates about an axis and images an object, we can observe the effect that the image of the object will not translate when the rotational axis is directly above the object. We use this effect as the basis of a simple algorithm for performing visual servoing which uses an approximation to the Image Jacobian to control the servoing process. This method exploits a control signal which is proportional to the error signal.

This work is characteristic of research currently examined in the domain of active vision, which includes moving cameras to obtain dynamic rather than static images and often includes the use of vision as a feedback signal for real-time control. Due to space constraints, we are unable to document the full gamut of literature pertaining to visual servoing; for a representative overview of the field, see Blake and Yuille [1].

2 OVERVIEW OF METHOD

Research in peg-in-hole servoing tasks is rich and varied. Some of the classical techniques include the work of Nevins and Whitney[8] in Remote-Center-Compliance (RCC) and the work of Lozano-Pérez et al.[7] with back-projections.

*This work was supported in part by DARPA contract DACA-76-92-C-007, NSF grants IRI-86-57151, CDA-90-24735, North American Philips Laboratories, Siemens Corporation and Rockwell International.

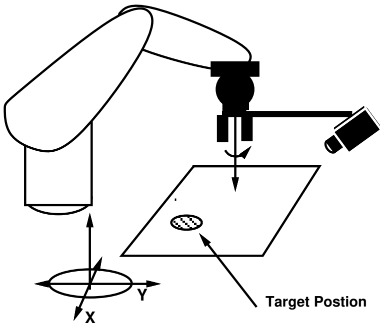

Figure 1: Experimental setup

These works try to solve the peg-in-hole problem primarily as a navigation problem (first, determining if a route exists from the peg to the hole, and second, determining what is the best path from the peg to the hole). Our work, on the other hand, only concerns itself with the task of aligning the end effector with the target. However, our work is extremely useful in solving the initial positioning problem for alignment and is consistent with the use of these previously mentioned methods.

A simplified schematic of our experimental setup is shown in figure 1 and figure 2 is an overhead view. The task of the vision-robot system is to insert the tip of the probe, which is mounted on the end-effector of the robot arm coincident with the robot's final axis, into one of the small holes in the block on the table in figure 2. The camera is mounted on the wrist of the end-effector such that it rotates about the final axis of the robot and images the area directly below the end-effector.

The method relies on the following simple effect: as we rotate the camera system about the robot's final axis (see figure 3), those objects which are farther away from the axis of rotation move a greater distance in the image plane than those objects closer to the axis. If we track these objects in the image plane during the rotation, they trace out elliptical paths. We refer to these ellipses as image-ellipses.

Figure 4 shows a series of simulated image-ellipses formed by tracking a feature point in the image plane. The family of image-ellipses was generated by changing the distance of the tracked feature point from the axis about which the camera system rotates. The center dot is an image-ellipse generated when the tracked feature point is coincident with the rotational axis (or in other words, when the manipulator is right over the point feature). As the tracked feature point is moved farther and farther away from the rotational axis, it sweeps out larger and larger image-ellipses (which satisfies our intuition).

In figure 5, the robot-camera system is constrained to move in the plane A which is defined by the circle swept by

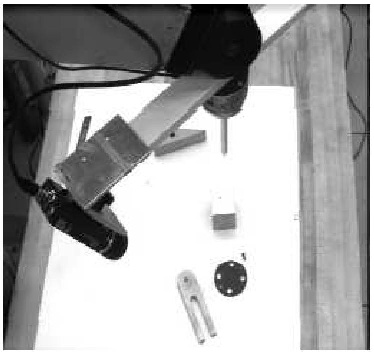

Figure 2: Overhead view of camera, robot, and multiple target setup

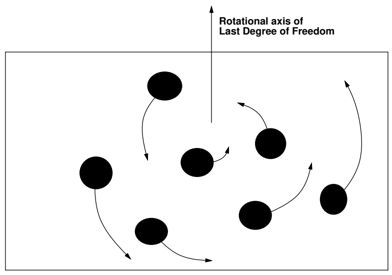

Figure 3: Image space movement of objects due to camera rotation about the final axis of the robot.

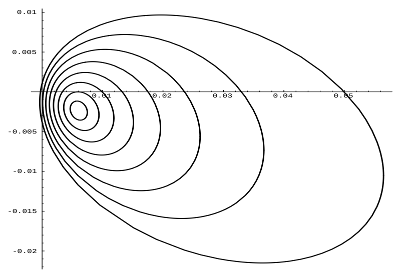

Figure 4: Family of image-ellipses generated by changing the distance of the tracked feature point from the axis about which the camera system rotates.

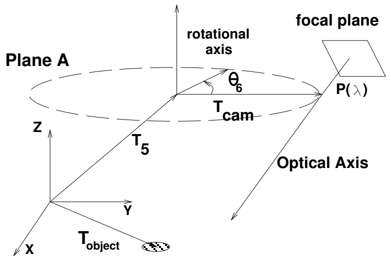

Figure 5: Overview of the coordinate systems.

the camera around the rotational axis. The camera subsystem itself is constrained to move in a circular orbit around the axis of the last joint of the robot. In order to align the peg with the hole, we examine the movement of a hole feature, which projects to a small set of pixels in image space. By moving the robot-camera system to positions in plane A, and rotating the camera system about the final axis, we can generate image-ellipses and compute their area. The heart of the method is creating a search strategy which minimizes the number of positions that the robot-camera system must move to before the alignment condition occurs. We broke the peg-in-hole task into two parts: the alignment task and the actual insertion task. The alignment task servos the end effector in a plane in robot space until the alignment condition occurs (when the peg and hole lie on the same axis). Once the alignment has been performed in plane A, the only movement necessary for insertion of the peg into the hole is a pure translation along the rotational axis (in our scenario, the Z-axis).

To perform our peg-in-hole insertions we also make the following assumptions (see figure 5):

- T_5, the transform from the world coordinate system to the end-effector of the robot, minus the last rotational degree of freedom, is known.
- R, the rotational transform for the final axis, θ_6, of the robot is known.
- T_{cam}, the camera transform matrix, is unknown.
- T_{object}, the transform from the world system to the feature point where the probe is to be inserted, is unknown.
- $P(\lambda)$, the perspective effect introduced by the camera system with focal length λ is unknown.
- Once the camera is mounted, it remains fixed with respect to the rotational axis of the final joint (6th) of the robot.
- The camera must be able to image the target object during any servoing operation. The object should never leave the focal plane.
- The object remains a fixed distance below the robot (we assume a constant Z value during the alignment operation).
- The features to be tracked are point-like features.

Intuitively, all we require is a robot that can tell us where its end-effector is and a camera system that rotates about the robot's final axis while keeping the feature point in view. We do not know where the camera is located with respect to the robot and we do not require the optical axis of the camera to intersect the final rotational axis of the robot. [1]

3 RECOVERY OF IMAGE-ELLIPSE PARAMETERS

A number of other researchers have created a body of literature on the recovery of ellipses from point data (see Safaee-Rad et al. [10] and Sawhney et al. [11].) These algorithms typically use data sampled from the full circumference of the ellipse (i.e. sampled over 2π radians). While these methods work very well, sampling over a complete rotation about the robot's axis is very slow, and precludes real-time use of these methods. Accordingly, our algorithm requires that we sample the ellipse only over a small part of a complete rotation. In our current setup, we sweep our camera system over $\pi/2$ radians in world space. [2]

As we sweep the camera system about the rotational axis, we accumulate sets of samples (θ_i, U_i, V_i) where θ_i is the rotational angle of the robot's final axis as the tracked feature point is imaged, and (U_i, V_i) are the coordinates of the feature point in camera space.

We then parameterize the curve traced out by the feature as:

$$U(\theta) = A\cos(\theta) + B\sin(\theta) + C \tag{1}$$

$$V(\theta) = D\cos(\theta) + E\sin(\theta) + F \tag{2}$$

[1] To see this, imagine the system setup (camera, extending rod, gripper and task system) where each piece is immovable. Notice that the rotational axis projects to a line in the camera imaging area. This line, by definition of the rigid system, cannot change. Its position is dictated by a fixed projection and since we use as our driving assumption the fact that the line simply rotates around its own symmetrical axis, its position does not change. Otherwise, the rotational axis shifts while it rotates. (This would mean that the rotational axis is not purely rotational and may actually have translational components which must be dealt with separately).

[2] The $\pi/2$ radian swath in world space does not usually correspond to an equal $\pi/2$ radian swath in camera space. Because of the perspective distortion, the path swept out in the camera system will either be smaller or larger than $\pi/2$ radian.

We fit these equations to the triplet data using linear least squares. Upon recovering the ellipse parameterization, we are able to compute the area enclosed by the ellipse by using the following formula:

$$Area = (A^2 + B^2 + D^2 + E^2)\frac{\pi}{2}. \qquad (3)$$

(which was derived using the previous two equations and Green's Theorem).

4 ESTIMATING THE IMAGE JACOBIAN

Our method to perform the alignment task revolves around the use of the Image Jacobian. The Image Jacobian has been used by a number of other researchers including Weiss et al. [13], Feddema et al.[4], Hashimoto et al.[5], Chaumette et al.[3], and Castaño et al.[2]. These methods track feature points and effect servoing movements using an Image Jacobian which relates Cartesian movements with positional errors derived from the tracked features. Other methods include the work of Papanikopolous et al. [9], Koivo et al. [6] and Miller [12].

The basic idea behind the Image Jacobian is to model the differential relationship between the camera system and the robotic control system in order to accurately predict the effects of small changes in one system on the other. It is a linear, position-dependent (i.e. non-constant) transform.

The relationship between the camera and world coordinate systems is given by the following formula:

$$w \begin{bmatrix} U_i \\ V_i \\ 1 \end{bmatrix} = \cdot P(\lambda) \cdot [T_5 \cdot R(\theta_6) \cdot T_{cam}]^{-1} \cdot \begin{bmatrix} X_{world} \\ Y_{world} \\ Z_{world} \\ 1 \end{bmatrix} \qquad (4)$$

Equation 4 can be differentiated with respect to time and the resulting equation is the differential relationship which relates the two coordinate systems. However, full knowledge of equation 4 is necessary to correctly derive the Jacobian matrix. In our system, since we assume that the calibration between the camera system and robot system is unknown, the task of recovering the actual Image Jacobian cannot be done directly.

We are only interested in control movements in the 2-D X-Y plane of the robot (we assume the alignment occurs at a fixed depth Z above the object). Accordingly, we can state the differential relationship between the camera coordinate system and the robot control system as follows:

$$\begin{bmatrix} \delta X \\ \delta Y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \delta U \\ \delta V \end{bmatrix} \qquad (5)$$

which basically means that small perturbations around some point in image space (U,V) can be linked to some small, linear move in world space (X,Y). The variables a, b, c, and d (the components of the Jacobian) are dependent on the robot parameters, the transformation between the camera system and the robot system, and the camera system parameters. However, we can estimate the Image Jacobian empirically using simple known movements in Cartesian space, and observing the movement of the object feature in camera space. By continuously updating the Image Jacobian, we can use its estimate to servo the robot to the correct alignment position even though we have not calibrated the two systems.

Our method is as follows:

1. The robot moves to some real world position (X, Y, Z).
2. We rotate the camera $\pi/2$ radians about the robot's final axis, tracking the feature point in image space and we calculate the area of the image-ellipse traced out.
3. At the midpoint of the rotation, $\pi/4$ radians, the robot freezes the rotation angle and translates to positions $(X + \delta X, Y, Z)$ and $(X, Y + \delta Y, Z)$. The change in image coordinates of the tracked object feature (δU, δV) is calculated after each positional change.
4. Each change in X-Y coordinates (δX,δY) results in a change in image coordinates (δU,δV) which allows us to empirically solve for the components of the Image Jacobian, a, b, c, and d, in equation 5.
5. The alignment condition occurs when the image-ellipse formed by tracking the object feature's path has zero area. We can compute the vector $(\delta U_{center}, \delta V_{center})$ from the position in the image where we calculated the Jacobian to the center of the projected ellipse. We estimate the alignment position in image space as the center of the image-ellipse (see figure 6).
6. We can now use our computed Image Jacobian to transform the vector $(\delta U_{center}, \delta V_{center})$ into a control movement in Cartesian space (δX, δY).
7. This procedure is repeated until the image-ellipse resolves either to a single point or to an ellipse of area $\leq 1.0\,pixel^2$.

This algorithm approaches continuous control since it estimates differential changes at each step. Although the actual computed movement is larger than a differential move, the vector $(\delta U_{center}, \delta V_{center})$ provides both an accurate direction of movement in image space as well as

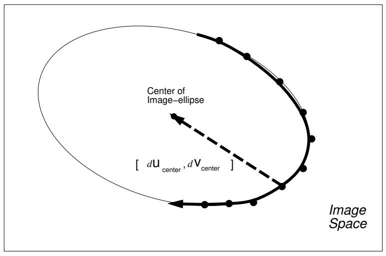

Figure 6: Estimating the image-space movement vector $(\delta U_{center}, \delta V_{center})$. This vector is derived from the point where the Image Jacobian was computed and the center of the image-ellipse. Using the Image Jacobian, $(\delta U_{center}, \delta V_{center})$ can be transformed into a Cartesian movement $(\delta X, \delta Y)$ to control the robot's movement.

a reasonable estimate of the magnitude of movement in Cartesian space so that we may limit overshoot and oscillation.

4.1 EXPERIMENT

In figure 1, we show a schematic of the system set up used for testing the new alignment method. We mounted a Sony XC-77 CCD camera in a bracket system off the end effector of a Puma 560 robot. The camera was not calibrated or position-constrained when initially placed. The robot system was controlled using RCCL. The images were digitized at 256x242 resolution and 8 bits gray scale at standard NTSC frame rates using the PIPE parallel image processing engine. The object was positioned so the robot would not encounter singularities when moving to new control positions. The other constraints on our system are listed in section 2.

At the beginning of the experiment, we performed the following operations:

1. The robot was moved to its initial X-Y position in plane A.

2. The user indicated the position in the image of the object feature to be tracked.

3. A moving edge-detection region was created about that point. The edge-detection region was used to track the object as the camera rotated about the robot's final axis.

We started the robot system at point $(0, 0)$ in Cartesian space (with the object approximately 10cm away in the X-Y plane). We ran the experiment using the Image Jacobian to control the movement of the robot system. The tracked feature was a 2mm diameter hole. Figure 9 shows the robot system placing a "peg" (the tapered probe) within 3mm of the center of the hole using uncalibrated camera data. The algorithm performed the same task over several different runs with about the same accuracy.

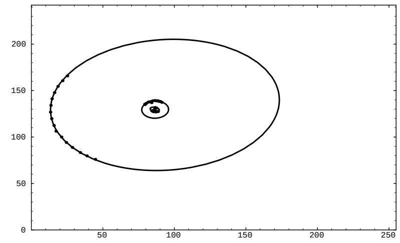

Figure 7: The image-ellipses formed by tracking the object over all 4 positions moved to by the robot.

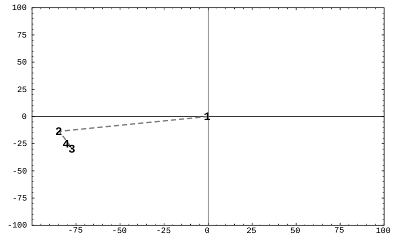

Figure 8: Positions examined by the Image Jacobian-based controller.

Figure 7 shows an overlay of all ellipses generated by the Image Jacobian control algorithm while trying to find the alignment condition. Notice that the system converges rapidly to the solution state. The estimate of the movement vector in image space $(\delta U_{center}, \delta V_{center})$ and the computed Image Jacobian provide accurate movements in Cartesian space to position the robot Probably the clearest indication of the Image Jacobian's convergence property is the result gained from plotting the positions moved to by the robot in Cartesian space (figure 8). It is possible to construct pathological situations where the algorithm may oscillate (by placing the camera system and object in positions where the center of the ellipse lies very close to one edge of the ellipse), but these cases are rare.

5 CONCLUSIONS

We have described a technique for performing an alignment task without using calibrated cameras. The technique exploits a simple geometric effect, rotational invariance, to perform the alignment. In our system, we mount a camera on the wrist of an end-effector such that it rotates about the final axis of a robot and images the area directly below the end-effector. By tracking the movement of the projection

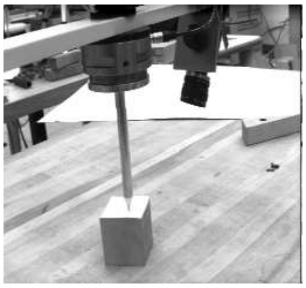

Figure 9: Robot system after performing the insertion task

of a point-like feature in image space, we can determine that the feature is aligned with the final axis when its projection simply rotates and does not translate during the rotation.

We have created a controller which uses rotational invariance and the Image Jacobian to effect active visual control of the alignment process. The key idea behind the Image Jacobian-based controller was to servo to the center of the image-ellipse (the best approximation to the point of rotational invariance) using a discrete approximation of the Jacobian at a given point. By exploiting an effect which is more continuous by nature, it was possible to create a servoing algorithm which converged smoothly and quickly to the alignment position. Using this controller, the robot system with an uncalibrated camera was able to place a peg within 3mm of the center of a 2mm hole with a high degree of repeatability.

This research indicates the possible existence of a whole class of servoing techniques which are orthogonal to calibration oriented methods. These techniques use simple geometric constraints to give robots the power to servo visually. While calibration is necessary and useful for many vision-robot tasks, we have shown that it is possible to use a non-calibrated technique to perform a task which was originally classified as needing calibrated vision. Also, by not using a calibrated camera, we do not suffer from the known errors of calibrated systems.

Our future research includes relaxing the constant Z constraint, the implementation of a faster image-space tracking system, and the investigation of other non-calibrated techniques for visual-servoing.

References

[1] A. Blake and A. Yuille, editors. *Active Vision*. MIT Press, 1992.

[2] A. Castano and S. Hutchinson. Visual compliance: Task-directed visual servo control. Technical Report UIUC-BI-AI-RCV-93-01, The Beckman Inst.: University of Illinois, 1993.

[3] F. Chaumette, P. Rives, and B. Espiau. Classification and realization of the different vision-based tasks. In K. Hashimoto, editor, *Visual Servoing*, pages 199–228. World Scientific, 1994.

[4] J. Feddema, C. S. G. Lee, and O. Mitchell. Weighted selection of image features for resolved rate visual feedback control. *IEEE Transactions on Robotics and Automation*, 7(1):31–47, Feb. 1991.

[5] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura. Manipulator control with image-based visual servo. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 2267–2271, 1991.

[6] A. Koivo and N. Houshangi. Real-time vision feedback for servoing robotic manipulator with self-tuning controller. *IEEE Transactions on System, Man, and Cybernetics*, 21, No. 1:134–142, Feb. 1991.

[7] T. Lozano-Perez, M. Mason, and R. Taylor. Automatic synthesis of fine-motion strategies for robots. *Int'l Journal of Robotics Research*, 3(1):3–24, 1984.

[8] S. Nevins and D. Whitney. Computer-controlled assembly. *Scientific American*, 238:62–74, February 1978.

[9] N. Papanikolopoulos, P. Khosla, and T. Kanade. Vision and control techniques for robotic visual tracking. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 857–864, April 7-12 1991.

[10] R. Safaee-Rad, I. Tchoukanov, K. C. Smith, and B. Benhabib. Three-dimensional location estimation of circular features for machine vision. *IEEE Transactions on Robotics and Automation*, 8:624–640, 1992.

[11] H. S. Sawhney. *Spatial and Temporal Grouping in the Interpretation of Image Motion*. PhD thesis, Univ. of Mass.-Amherst, Feb 1992.

[12] I. W. Thomas Miller. Sensor-based control of robotic manipulators using a general learning algorithm. *IEEE Journal of Robotics and Automation*, RA-3(2):157–165, April 1987.

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