Some Problems in Adaptive Visual Servoing
or
How to Keep the Left Eye From Knowing what the Right is Doing

Gregory D. Hager

Research Report YALEU/DCS/RR-948
January 1993

Some Problems in Adaptive Visual Servoing
or
How to Keep the Left Eye From Knowing What Right is Doing

Gregory D. Hager
Dept. of Computer Science
Yale University
P.O. Box 2158 Yale Station
New Haven, CT 06520

WORKING PAPER
January 28, 1993

Abstract

In this report, we describe how various visual servoing problems can be implemented without explicit knowledge of hand-eye calibration or calibration between two stereo cameras. The results seem to indicate that it is possible to perform many visually-guided tasks using minimal calibration information.

1 Introduction

This paper discusses the basic ideas behind *image encodeable tasks.* The motivation behind image encodeable tasks is to avoid sensitivity to hand-eye calibration in visual servoing systems. The basic methodology is twofold:

- Describe control objectives in terms of the visual observables that are independent of many parameters of the hand-eye system.
- Use adaptive control techniques to perform online correction of any remaining calibration parameters.

In this short report, we present two examples to illustrate the basic concepts. The first example falls squarely in the realm of classical linear adaptive control, and serves to illustrate the methodology. The second problem leads to a nonlinear adaptive control problem. We then discuss what appear to be useful visual stategies which rely on exploiting certain special visual structures. These strategies lead to a simplified form of the control problem which appears to be more tractable. At the time of this writing we are looking for solutions to the remaining adaptive control issues and implementing our results.

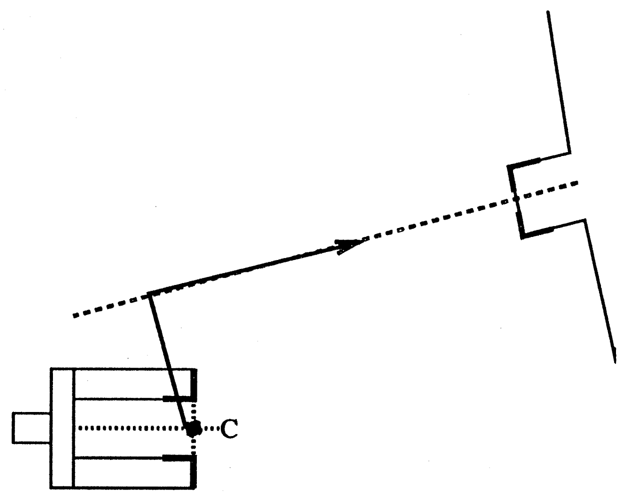

Figure 1. The geometry of the two dimensional visual control problem.

2 A Linear Adaptive Problem

To illustrate the basic mechanisms of adaptation that we are proposing to use, consider the following problem.

Example 2.1 Suppose that a robot manipulator is only allowed to move in the plane parallel to the imaging plane of an observing camera. Referring to Figure 1, the dark solid lines are the tracked features on the gripper and on a peg-like fixture onto which the gripper is to be guided. Furthermore, assume that the orientation of the camera system about the optical axis is approximately known, either though observing the gripper and comparing its observed attitude to the robot's estimate, or via an offline calibration. The distance from the camera to the plane of robot motion is unknown. The problem is to guide the gripper onto the peg.

If manipulator motions are commanded relative to the endpoint of the gripper (marked C in the diagram), there are two independent regulation problems: aligning the gripper with the peg, and moving it along the central axis of the peg until it contacts the surface. The problem of alignment is straightforward: by measuring the difference in angles of the gripper and the peg, the gripper can be rotated about its endpoint until alignment occurs. Note that problem is independent of any calibration information and can be solved using a standard linear regulator.

The regulation of translation is more difficult due to the unknown scale factor between commanded and observed displacement, and the possibility of errors in the estimated orientation of the camera coordinate frame. This problem can be formulated as a continuous time adaptive estimation problem as follows [2]:

$$
\begin{aligned}
\dot{x}_p(t) &= u_p(t) \quad \text{(plant)} \\
\dot{x}_m(t) &= kRu_m(t) \quad \text{(model)}
\end{aligned}
$$

The constant k describes the perspective scale factor and R is a rotation matrix. These factors can be combined into a single 2 by 2 matrix expressed as a 4 element column vector M. Define the error between the output of the model and the observed output plant to be $e(t) = x_p(t) - x_m(t)$, define I_2 to be the 2 by 2 identity matrix, and let $U(t) = u(t)^T \otimes I_2$ where

$$[a, b] \otimes I_2 = \begin{bmatrix} a & 0 & b & 0 \\ 0 & a & 0 & b \end{bmatrix}.$$

By observing a sufficiently rich set of plant trajectories the adaptive law

$$\hat{M}(t+1) = \hat{M}(t) - U(t)^T e(t)$$

will cause $\hat{M}$ to converge asymptotically to M. Note that the problem is overparametrized—we are recovering 4 parameters instead of the original two.

In order to control the system, we reverse the role of the plant and model, again combine the scale and rotation factors, and introduce independent control parameter $r(t)$:

$$\begin{aligned} \dot{x}_m(t) &= M u_m(t) \quad \text{(plant')} \\ \dot{x}_p(t) &= r_p(t) \quad \text{(model')} \end{aligned}$$

Equating $u(t)$ and $r(t)$ and applying the adaptive law

$$\hat{M}(t+1) = \hat{M}(t) - R(t)^T e(t)$$

where $e(t) = x_p(t) - x_m(t)$ and $R(t) = r(t)^T \otimes I_2$ yields an adaptive system with the following properties:

1. For a given set of trajectories, the control error will converge to 0.
2. If the set of trajectories is "rich enough," the value of $\hat{M}$ will converge to M.

Figure 2 shows the results of a simulation using circular trajectories. We see that the output error is driven to zero for all cases where the initial estimate was close to the true plant parameters. The parameter error in the right graph shows that the matrix M is not always recovered correctly, though in most cases M is close enough to correct to drive the control error to zero. Due to the discrete time nature of the simulation, one of the cases examined did not converge.

Let p_g be the observed position of the gripper center (by differencing observations of the two jaws), p_p be the observed position of the center of the peg, θ_0 be the estimated position orientation of the camera coordinate frame relative to the robot coordinate frame, and α be the estimated orientation of the peg. We will generate a trajectory using the following function:

$$\begin{aligned} d &= p_p - p_g \\ G(d) &= \left[\min(g_1 d_x, 0.1)/(1 + g_2 d_y), \min(g_3 d_y, 0.1) \right]^T, \quad g_1, g_2, g_3 > 0 \\ \Delta r(t) &= R(\alpha) G(R(-\alpha) d) \end{aligned}$$

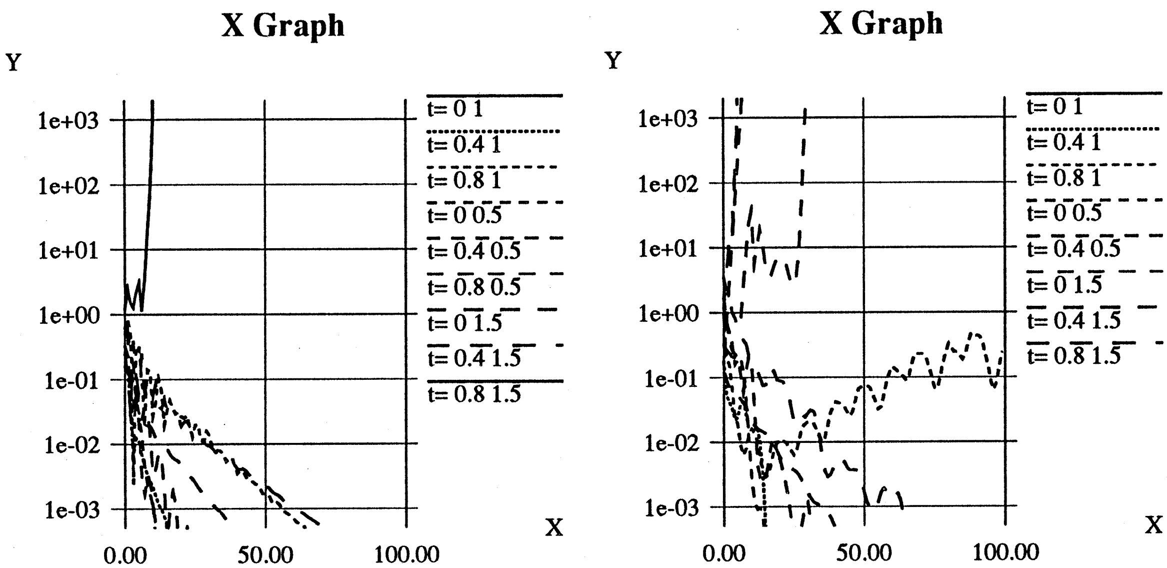

Figure 2. Left, the output error as a function of iteration; and right, the parameter error as a function of iterations.

This system is designed to emphasize alignment motions followed by approaching motions with different choices of the gain coefficients governing the speed of approach and the adherence to alignment with the axis of the peg.

Figure 4 illustrates the trajectories taken by the system (top) and the output and parameter errors (bottom) for a variety of different parameter settings as the system executes the approach task. The parameters used were $g_1 = 0.1, g_2 = 10.0$, and $g_3 = 0.5$. From the data, we see that:

- The task was successfully executed for all but one parameter setting. Again, this appears to be due to the effects of discretization and can be dealt with by damping the system slightly.
- The output error is quickly driven to zero.
- The parameter error is not driven to zero because the trajectories are not comprehensive enough to successfully recover all parameters.

This example illustrates the basic properties of our approach:

1. Visually encoded tasks reduce the amount of hand-eye calibration required to perform regulation. In the example above, only two hand-eye calibration parameters, k and θ are needed to perform the task even though four parameters are needed to define the full hand-eye relationship.

2. Independent visual and kinesthetic measurement of robot motion permits refinement of the remaining parameters of the hand-eye model while performing a task. Furthermore, these parameters are only "estimated" as needed.

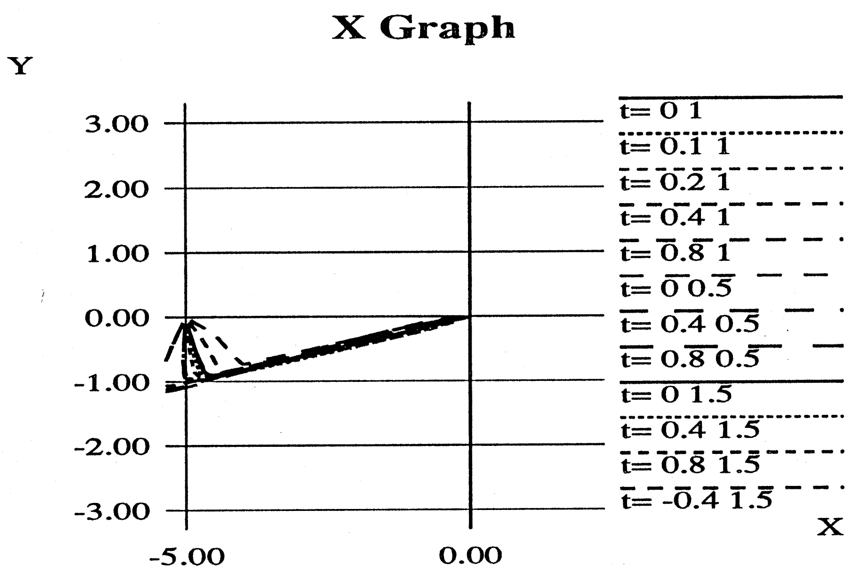

Figure 3. The trajectories of the system as a function of iterations.

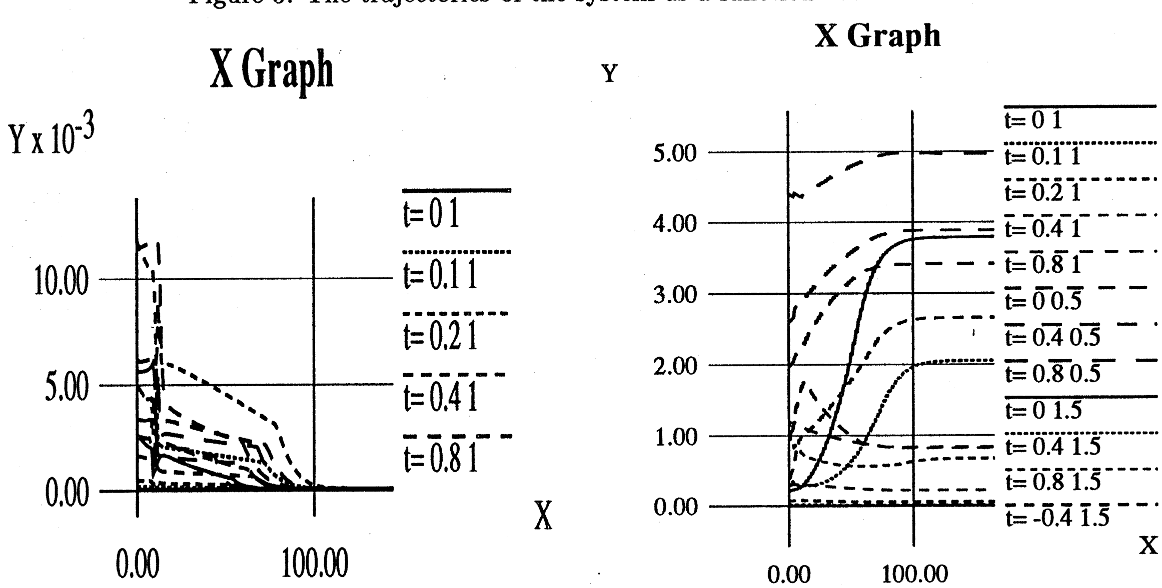

Figure 4. Left, the output error as a function of iteration; and right, the parameter error as a function of iterations.

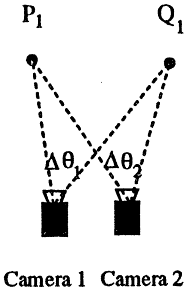

Figure 5. The basic stereo positioning task.

3 A NonLinear Adaptive Problem

In the previous section, we described a simple case where a manipulation system is controlled using a monocular camera. Given complete structural knowledge about what is observed and a sufficient number of observable features, it is sometimes possible to solve six-degree of freedom manipulation problems using only monocular cues. However, cases will arise where monocular cues do not suffice for stable, accurate control. Here, we show how it is possible to perform visual servoing using an *uncalibrated* stereo pair of cameras. The basic approach is illustrated for a planar problem in Figure 5. Let $P = (x, y)$ represent a point on a controlled system, and $Q = (u, v)$ represent a goal point. Define $\triangle\theta_1$ to be the angular difference in line-of-sight to points $P = (x, y)$ and $Q = (u, v)$ in camera 1, and $\triangle\theta_2$ be the same angular difference in camera 2. We then note that $\triangle\theta_1$ and $\triangle\theta_2$ are independent of camera orientation in the plane. Furthermore, given a nonzero baseline, b, and assuming that P and Q do not fall on the line through the optical centers of the cameras, $\triangle\theta_1$ and $\triangle\theta_2$ are zero if and only if $P = Q$. Hence, a regulator that reduces $\triangle\theta_1$ and $\triangle\theta_2$ to zero will move point P to point Q. Furthermore, the independence of $\triangle\theta_1$ and $\triangle\theta_2$ from camera orientation indicates that changes in pan-tilt angle to adjust field of view do not affect control in any way.

For the moment, let us assume that the origin of the hand-eye system is located midway between the optical centers of the cameras, the baseline length is $b = 2d$, and the manipulator is a perfect velocity controlled system with state $x = (x_1, x_2)$ and control input $u = (\dot{x_1}, \dot{x_2})$, both expressed in camera coordinates. We do not know $R \in SO(2)$, the relative orientation of the robot coordinate system. This allows us to describe the problem as a linear system with a nonlinear output map as follows:

$$\begin{aligned} \dot{x} &= Ru \\ \theta = \begin{bmatrix} \theta_l \\ \theta_r \end{bmatrix} &= \begin{bmatrix} \arctan\left(\frac{x_1 - d}{x_2}\right) - \alpha_l \\ \arctan\left(\frac{x_1 + d}{x_2}\right) - \alpha_r \end{bmatrix} \end{aligned}$$

where α_l and α_r are the pan angles of the two cameras.

We could try to get this into state-accessible form as we did in the previous section. Taking

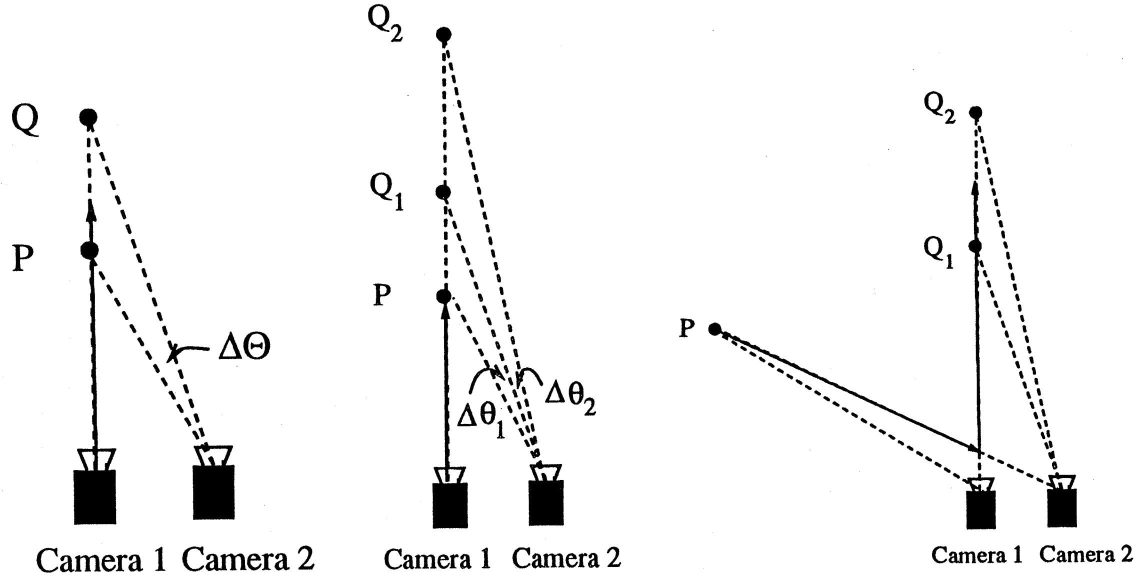

Figure 6. Positioning Tasks

derivatives yields:

$$\begin{bmatrix} \dot{\theta}_l \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \frac{\dot{x}y-(x-d)\dot{y}}{y^2+(x-b)^2} \\ \frac{\dot{x}y-(x+d)\dot{y}}{y^2+(x+b)^2} \end{bmatrix}$$

From this, we see that the direct relationship between $\dot{\theta}$ and u is nonlinear, and probably not solvable using any standard techniques.

3.1 Simplifying the Problem

Although the general qualitative stereo control problem falls outside the realm of "standard" adaptive control, there are specific cases that look more promising. Specifically, observe that, while qualitative stereo in theory provides the means for reaching goal positions without detailed hand-eye calibration information, it does not allow much control over the *cartesian path* of the approach. Hutchinsen's notion of a visual constraint surface [1] allows us to extend this idea to control path of approach, and to perform a variety of more complex positioning tasks. Figure 6 describes a few of the possibilities. Figure 6(a) shows how direction of approach can be controlled. One camera, hereafter referred to as the *dominant camera*, is moved to point along the direction of approach; it effectively holds the manipulator to the line of sight. The other camera, hereafter referred to as the *secondary* camera, can be used to compute visual distances to the target and control the speed of approach. This is effectively the strategy used by human when they try to do an accurate approach task such as threading a needle. Figure 6(b) shows how a relative positioning task can be accomplished. Again, by using camera 1 as an alignment tool, camera 2 can be used to compute visual angles determining the angles from P to Q_1 and Q_2. When the magnitude of the visual angle between P and both Q_1 and Q_2 is the same, the point P is between the two goal points.

Finally Figure 6(c) illustrates how more complex motions can be accomplished by trading between dominant cameras. First a motion along a ray emanating from the camera 2, is performed until the manipulator "contacts" the line joining the focal point of camera 1 and the target. Camera 1 becomes the dominant camera and the motion proceeds as in Figure 6(b).

When manipulator motion is constrained to rays emanating from the dominant camera the regulation problem takes a simpler form. In the dominant camera, the objective is to ensure that the manipulator moves so that the angle between the controlled system and a reference point *remains constant.* Conversely, the secondary camera seeks to bring the system to a halt *when it crosses a reference line of sight.* The control problem can now be partitioned into two distinct components:

Dominant Problem

The dominant camera system will control the *direction* of the control vector u. The goal is to choose u so that the controlled system moves along the designed ray through the origin of the camera system. The dynamical system with the dominant observer can be described in the form:

$$\begin{aligned} \dot{x} &= Ru, \quad R \in SO(2) \\ y &= \arctan(x_2/x_1) \end{aligned}$$

The control objective is to maintain $y = y_g$ where y_g is a given reference angle. The matrix R is an unknown parameter as is the initial value of the state vector x.

At the time of this writing, we do not know the solution to this control problem. However, its simple structure leads use to believe a solution is feasible.

Secondary Problem

The secondary camera system will control the *magnitude* of the control vector u. Its structural description is identical to the dominant camera (note that it has its own separate R and y_g parameters). Setting $||u|| = |y - y_g|$ will lead to a stable, convergent control in the continuous case.

The examples presented above can be carried out by the appropriate switching between the roles of the two observing cameras.

3.2 The 3D Problem

In practice, this form of visual servoing will need to take place in standard three-dimensional Cartesian space. When moving to 3D, the problem formulation doesn't change much, but some changes need to be made. First, let us assume that the goal is as above, namely to move the controlled system along a visual ray toward a goal point. Then we can formulate the problems as

Dominant Problem

The dominant camera system will again control the *direction* of the control vector $u \in \Re^3$. The goal is to choose u so that the controlled system moves along the designed ray through the origin of the camera system. The dynamical system with the dominant observer can be described in the form:

$$\begin{aligned} \dot{x} &= Ru, \quad R \in SO(3) \\ y &= \begin{bmatrix} \arctan(x_2/x_1) \\ \arctan(x_3/x_1) \end{bmatrix} \end{aligned}$$

The control objective is to maintain $y = y_g$ for some reference value y_g.The matrix R is an unknown parameter as is the initial value of the state vector x.

At the time of this writing, we do not know the solution to this control problem. Moreover, it is somewhat less likely to have a simple solution due to the complexity of the $SO(3)$.

Secondary Problem

The secondary camera system will again control the *magnitude* of the control vector u. As before, its structural description is the same as the dominant camera, however choosing its control law is slightly more difficult. However, if we assume that the x_3 coordinate axis of both cameras is nearly aligned (as is the case in most stereo systems), then setting $||u|| = |y_1|$ will lead to a stable, convergent control in the continuous case.

Now, consider the slightly more complex problem of moving the controlled system from an arbitrary point in space *to* a designated ray r in the secondary camera. The problem is that the secondary camera must control some of the directional components since otherwise we can do no better than guarantee bringing the system to some point in the plane defined by r and the x_3 coordinate of the secondary system. We will again simply the problem by assuming that the x_3 axes of both cameras is aligned. Then the solution is:

Dominant Problem

The dominant camera system will now control the direction of the first two components of the control vector $u \in \Re^3$. The structure of the system is the same as that described above. Now, the control objective is to maintain $y_1 = y_g$ for some reference value y_g.

Secondary Problem

The secondary camera system will now control the *magnitude* of the control vector u as well as its third component. We will choose $u_3 = -y_2$ and then $||u|| = |y_1|$. We believe (we have not proved) that this will lead to a stable, convergent controller in the continuous case.

4 Conclusions

We have presented two visual servoing problems and examined their solutions using adaptive control techniques. Both examples involve control in the plane, although there is some hope that the ideas of the last section extend to 3-D problems. We are currently experimenting with heuristic solutions to the unresolved adaptive control issues, and are seeking to implement the techniques on our hand-eye system.

Acknowledgements

Dan Koditschek was instrumental is describing enough adaptive control so that I could pose the basic problems described here, and he also suggested the title "image encodable tasks."

This research was supported by DARPA grant N00014-91-J-1577, by National Science Foundation grants IRI-9109116 and DDM-9112458, and by funds provided by Yale University.

References

[1] S. Hutchinson. Exploiting visual constraints in robot motion planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1722–1727. 1991.

[2] K. Narendra and A. M. Annaswamy. *Stable Adaptive Systems.* Prentice Hall, Englewood Cliffs, N.J., 1989.