An Active Visual Estimator for Dexterous Manipulation

A. A. Rizzi [*]and D. E. Koditschek [†]

Artificial Intelligence Laboratory, University of Michigan,
Department of Electrical Engineering and Computer Science

Technical Report CSE-TR-213-94

Prepared for presentation at the 1994 IEEE International
Conference on Robotics and Automation
Workshop on Visual Servoing

Abstract

We present a working implementation of a dynamics based architecture for visual sensing. This architecture provides field rate estimates of the positions and velocities of two independent falling balls in the face of repeated visual occlusions and departures from field of view. The practical success of this system can be attributed to the feedback interconnection between two strongly nonlinear dynamical systems: a novel "triangulating" state estimator; and an image plane window controller. We detail the architecture of this active sensor, provide data documenting its performance, and offer an initial analysis of its soundness in the form of a convergence proof for (a simpler version of) the estimator and a boundedness proof for (a somewhat idealized version of) the manager.

1 Introduction

We have built a three degree of freedom robot that bats two balls into simultaneous stable periodic vertical trajectories that commonly persist for the better part of an hour [20]. The juggling algorithm underlying this behavior relies on a continuous stream of ball position and velocity estimates delivered by a stereo camera system that views brightly illuminated white balls against a dark background. Despite the structured visual environment, this system did

[*] Supported in part by IBM through a Manufacturing Graduate Fellowship and in part by the National Science Foundation under grant IRI-9123266.

[†] Supported by the National Science Foundation in part through a Presidential Young Investigator Award and in part under grant IRI-9123266.

not work until we had replaced our originally conceived sensor subsystem — essentially a linear observer driven by image plane centroids — with a more complicated scheme that feeds back the state of a nonlinear "triangulating observer" to be used in the control of a "camera window manager." At a time when many in the robot vision community are exploring the benefits of "visual servoing" or have found the need for including "attention mechanisms" in their camera architectures, we offer this account as documentation of a particular system which seems to incorporate most of the essential features of the "active sensor" yet remains simple enough to permit some formal analysis.

Developing and reasoning formally about this specific system interests us more generally in view of the apparent need to develop a theory and practice of "dexterous robots." This term, as we understand it, denotes an autonomous machine capable of interacting with a dynamical world. The strategies of general interest to us are, as in the present case, feedback algorithms: they specify the manipulator's actions at each instant in time as a function of its current state and that of the world. For a juggling machine, the world's state reduces to the current position and velocity of one or two balls and the task of estimating this state forms the narrow focus of the present paper. It is our belief that a much larger range of dynamically dexterous tasks (of which juggling is but a simple example) will necessitate the ability to generate timely and accurate estimates for the state of the environment independent of the specific control algorithm.

Of necessity such estimation will require a model of the world's dynamics. This estimation model will necessarily include both a set of state variables (which describe the system's belief about the current condition of the world) and a set of parameters (which describe the behavior of the world, or possibly how we observe the world). It follows that there are two classes of "dynamical vision" problems and two forms of each class to be considered. One may attempt to control or mearly estimate the world state variables. Either class of problem may in turn take form with or without prior knowledge of the exact parameters. Several of these four logical variations have been well studied in the literature [16, 1, 11, 6]. The work presented here focuses on what may be arguably be the simplest: the problem of estimating the state of the world given explicit prior knowledge of the relevant parameters. Related research, often categorized as *visual servoing*, normally focuses on the direct control of the world state given similar parametric information [15]. Recently there has emerged a body of work focused on this same control problem, but without the presumption of complete prior calibration [11]. We will offer more comments below on the relationship between these problems.

Given the desire to perform tasks in the physical world the natural sensor choice would provide cartesian measurements. However cartesian sensors are expensive and often complex. One common alternative is to employ a stereo vision system of some sort (e.g. two cameras, one camera with constrained objects, one camera and structured lighting, etc.). Of course these sensors are nor truly cartesian: they actually provide a non-linear measurements of the current configuration of the world state. The signal processing required to interpret these measurements forms the topic of this paper. Beyond the description in Section 2 of a successful laboratory architecture, the paper presents two separate but interrelated contributions. Section 4 examines the details of our approach to the "active vision" problem, the underlying goal being to develop an analytic understanding of reliable strategies capable of managing the acquisition of sensor data in such a way that both estimator convergence and future measurement acquisition can be guaranteed. Section 5 presents a new approach to state estimation based on visual data, which makes use of a dynamic filter to perform triangulation in much the same way "visual servoing" makes direct use of visual data to achieve task level goals. The desired measurements are normally computed from the sensor data through inversion of the sensing operation (i.e. triangulation), whereas we describe a new approach to this inversion problem based on a combination of estimator theory and non-linear least squares optimization. Our experience suggests that the construction of actual system capable of dexterous manipulation will require the inclusion of both active vision and estimation subsystems. It is this inevitable combination, and the need to understand the implications of their interaction that motivates our addressing both topics here.

2 Setting

The class of sensor systems we wish to build require an understanding both of how the world they attempt to sense evolves over time, and how they perceive that world. Thus we pause here to develop simple models for the falling and bouncing ball, the robot juggling strategy, and the robot's physical sensors (each will be used repeatedly below).

2.1 Physical Models: The Robot's Environment

State estimation can only be as effective as the model of the environment. For the juggling problem, the model in question will consist of two parts: flight, which describes the behavior of a ball under the influence of gravity; and impact, which describes how a ball will bounce off the robot's paddle.

2.1.1 Flight Model

For simplicity, we have chosen to model the ball's flight dynamics as a point mass under the influence of gravity. A position-time-sampled measurement of this dynamical system will be described by the discrete dynamics,

$$\begin{aligned} w_{j+1} &= F^s(w_j) \triangleq A_s w_j + a_s; \\ & A_s \triangleq \begin{bmatrix} I & sI \\ 0 & I \end{bmatrix}; \quad a_s \triangleq \begin{bmatrix} \frac{1}{2}s^2\tilde{a} \\ s\tilde{a} \end{bmatrix} \\ b_j &= Cw_j; \qquad C = [I, 0], \end{aligned} \tag{1}$$

where s denotes the sampling period, $\tilde{a}$ is the gravitational acceleration vector, and $w_j \in I\!R^6$ embodies the entire state of the object (its position and velocity).

2.1.2 Impact Model

Consider a ball with trajectory $b(t)$ colliding with the paddle in robot configuration $q \in \mathcal{Q}$ (depicted in Figure 1) at some point, p on the paddle which has a linear velocity v. We seek a description of how the ball's phase, $(b, \dot{b})$, is changed by the robot's phase, $(q, \dot{q})$, at an impact.

As in [5, 13, 19] we will assume that the components of the ball's velocity tangent to the paddle at the instant of contact are unchanged by impact, while the change in the normal component is governed by the simplistic (but standard [23]) coefficient of restitution law. For some $\alpha \in [0,1]$ this impact model can be expressed as $(\dot{b}'_n - v'_n) = -\alpha(\dot{b}_n - v_n)$, where $\dot{b}'_n$ and v'_n denote the normal components of the ball and paddle velocities immediately after impact, while $\dot{b}_n$ and v_n are the velocities prior to impact. Assuming that the paddle is much more massive than the ball (or that the robot has large torques at its disposal), we conclude that the velocity of the paddle will remain constant throughout the impact ($v' = v$). It then follows that the coefficient of restitution law can be re-written as $\dot{b}'_n = \dot{b}_n + (1+\alpha)(v_n - \dot{b}_n)$. and, hence,

$$\dot{b}' = \dot{b} + (1+\alpha)nn^T(v - \dot{b}), \tag{2}$$

where n denotes the unit normal vector to the paddle.

2.2 Behavioral Model: The Robot's Strategy

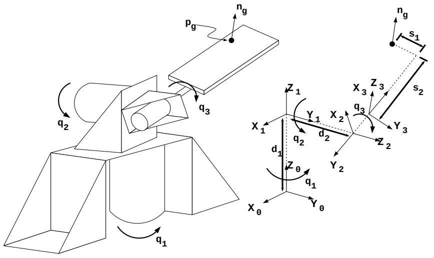


Figure 1: The Bühgler arm (left) and it's kinematics (right) [22].

A detailed development of our juggling control strategy can be found in [17]. Briefly, the "mirror law," is a map (m) from the phase space of a ball into the configuration space of the robot. Thus the robot's reference trajectory is specified by $q_d(t) = m(w(t))$, where $w(t)$ denotes the state of the ball. The function, m is defined as follows. Using (6) from [21], define the joint space position of the ball

$$\begin{bmatrix} \phi_b \\ \theta_b \\ \psi_b \\ s_b \end{bmatrix} \triangleq g^{-1}(b), \tag{3}$$

where g^{-1} is the inverse kinematic map (including the paddle's length s that provides an effective fourth degree of freedom) for our machine, which is shown in Figure 1. We now express formulaically a robot strategy that causes the paddle to respond to the motions of the ball in four ways:

(i) $q_{d1} = \phi_b$ causes the paddle to track under the ball at all times.

(ii) The paddle "mirrors" the vertical motion of the ball through the action of θ_b on q_{d2} as expressed by the original planar mirror law [5].

(iii) Radial motion of the ball causes the paddle to raise and lower, resulting in the normal being adjusted to correct for radial deviation in the ball position.

(iv) Lateral motion of the ball causes the paddle to roll, again adjusting the normal so as to correct for lateral position errors.

To this end, define the ball's *vertical energy* and *radial distance* as

$$\eta \triangleq \gamma b_z + \frac{1}{2}\dot{b}_z^2 \quad \textit{and}, \quad \rho_b \triangleq \sin(\theta_b) s_b \tag{4}$$

respectively. The complete mirror law combines these two measures with a set point description ($\bar{\eta}$, $\bar{\rho}$, and $\bar{\phi}$) to form the function

$$m(w) \triangleq \begin{bmatrix} \overbrace{\phi_b}^{(i)} \\ \underbrace{-\frac{\pi}{2} - (\kappa_0 + \kappa_1(\eta - \bar{\eta}))\left(\theta_b + \frac{\pi}{2}\right)}_{(ii)} \quad + \\ \underbrace{\kappa_{00}(\rho_b - \bar{\rho}_b) + \kappa_{01}\dot{\rho}_b}_{(iii)} \\ \underbrace{\kappa_{10}(\phi_b - \bar{\phi}_b) + \kappa_{11}\dot{\phi}_b}_{(iv)} \end{bmatrix}. \tag{5}$$

The important idea to note is that the "strategy" we have chosen to implement presumes continuous availability of the ball's position and velocity. As more systems are designed to function in dynamic settings it seems reasonable to expect more and more systems to require such information about their environment .

2.3 Sensing Model: The Robot's Sensors

For the juggling system, the focus of this work, the available data consists of two fields of image data which are simultaneously acquired from two cameras. It is then the responsibility of the sensing system to report the location of the ball (or balls) in space. As stated above, the visual environment is structured such that an individual pixel may be identified as either part of a ball or the background simply by its intensity – we are looking for white balls against a black background. This structure allows us to use a "simplistic" geometric model of the world (pixels are either part of the ball or not) to simplify the image processing. Although we have chosen to make use of structured lighting, the environment is far from uniform. As a ball travels across the image it appears to change shape due to the lighting effects. Thus a geometry based vision system could reliably report ball locations only if it was capable of taking into account these poorly modeled lighting effects. As will be seen we use the dynamic model to make up for this lack of geometric detail.

2.3.1 Camera Model

The simple projective stereo camera model of the form,

$$c : I\!R^3 \rightarrow I\!R^4,$$

(which maps positions in affine 3-space to a pair of image plane projections in the standard manner) has been sufficient for the experiments associated with this paper. Knowledge of the cameras' relative positions and orientations together with knowledge of each camera's lens characteristics (at present we model only the focal length) permits the selection of a "pseudo-inverse" or "triangulation-function,"

$$c^{\dagger} : I\!R^4 \rightarrow I\!R^3, \tag{6}$$

such that $c^{\dagger} \circ c = id_{I\!R^3}$. We have discussed our choice of pseudo-inverse at length in previous publications [21], and details of the calibration scheme can be found in Appendix A and [19].

More precisely, c, is formed by stacking together the perspective projections due to the two individual cameras,

$$v := c(b) = \begin{bmatrix} \pi_{f_1}\,(\,{}^1\!H_0\, b) \\ \pi_{f_2}\,(\,{}^2\!H_0\, b) \end{bmatrix}, \tag{7}$$

where

$$\pi_f(b) := \frac{f}{b_3}\begin{bmatrix} b_1 \\ b_2 \end{bmatrix},$$

${}^i\!H_0$ is the the homogeneous-matrix representation of the base frame in the ith camera's frame, and f_i is the ith camera's focal length.

2.3.2 Impact Detection

One significant drawback of a camera system as the primary sensor is its relatively low data rate: determining the exact time that a falling ball impacted a moving surface from a 60Hz data stream is difficult. In order to implement an estimation system capable of dealing with such events we require both a model of impact (presented above) and rather precise knowledge of the time the impact takes place. To detect the impacts we have chosen to augment the sensing system with a physical *impact detector.* This device consists of a single microphone attached directly to the robot paddle whose output is passed through a narrow band filter tuned to the fundamental frequency produced by the impact, then rectified and threshold detected.

3 An Active Visual Estimator

The design of a complete sensing system for an environment such as that just presented requires the careful integration of a number of functional submodules. In the following section we attempt to explore both our experience constructing such a system as well as the manner in which those experiences have lead to an architectural framework suitable for its analysis. Note that although we dedicate much of this section to the development of this particular architecture, we hope to suggest a framework suitable for the analysis and interpretation of dynamical sensor systems in general. We conclude this section with experimental results gathered from the juggler's sensor system in order to highlight the benefits of the dynamical sensing framework.

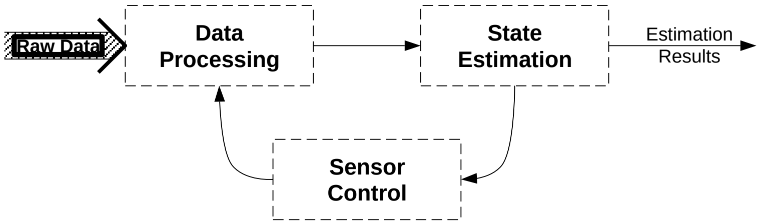


Figure 2: A generic active visual estimator.

3.1 The Challenge of Constructing an Integrated System

Ideally a purely cartesian sensor could be purchased or designed which would be capable of providing the "continuous" state estimates necessary for implementation of a controller of the type described in Section 2.2. Currently, however, such sensors are either prohibitively priced or lack sufficient sophistication to cope with anything but the most stringently structured environment. We thus face the task of designing our own sensing system.

It seems natural to partition a sensing system suitable for this type of dynamical task into three subsystems as shown in Figure 2. This architecture separates the sensing system into the following modules: *data processing* encompasses the algebraic (memoryless) signal processing; *state estimation* contains the dynamic model based processing; and finally, *sensor control* implements the feedback segment responsible for guiding the "attention" of the low level data processing. Depending on the specific nature of both the sensors and the problem, any of the three abstract modules shown might become trivial. However it is our contention that this architecture can be found in nearly any system, and that thinking about the overall behavior in terms of these separate modules is advantageous.

The *sensor control* block in Figure 2 presumes a fundamental need and/or advantage to constructing "active" sensing systems. Our experimental experience and that of others [8, 7, 15] suggests that such an advantage both exists and can be practically exploited. Clearly, an active sensing system can be used to minimize the total incoming data by "focusing the attention" of the machine only where meaningful data is likely to be found. This type of improvement, however, seems at most superficial since if we had available sufficiently greater processing power the consideration would not arise. More importantly, "focusing the attention" of a machine can be used as a means to introduce temporal knowledge about the environment's behavior back into the data processing task, thereby making the "feature extraction" task more tractable. The simplest example of this can be seen in common approaches to solving visual correspondence problems in image sequences, where the presumed dynamic model for the world is either zeroth or first order (static or constant velocity) and features are matched based on their proximity to previous observations. There is a clear advantage in offering location clues to low level feature extraction, for if they include certainty bounds they may be used to perform an initial data segmentation, thus significantly simplifying the "early vision" problem.

3.1.1 An Initial Design

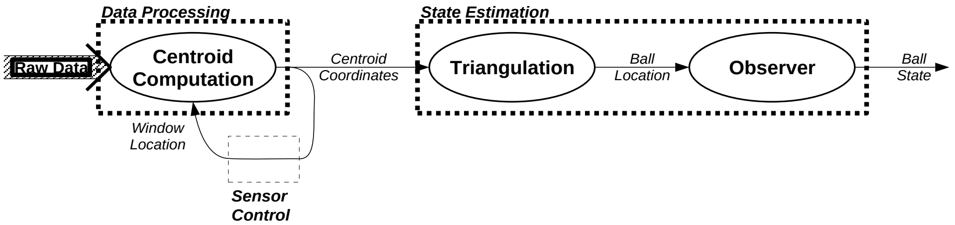


Figure 3: Our initial implementation of the active visual estimator.

Our initial foray into the design of a sensor system suitable for use in robot juggling followed basic engineering principles for reliable *signal processing* and *state estimation* modules. However, little thought was given to their interconnection. Consequently, the associated *sensor control* module connecting them was correspondingly trivial. Figure 3 depicts the architecture of this initial version of the system:

Data Processing Following Andersson's experience in real-time visual servoing [3] we chose to employ a first order moment computation applied to a small window of a threshold-sampled (thus, binary valued) image of each field. Thresholding, of course, presumes a visually structured environment. In our case white ping-pong balls are illuminated with halogen lamps while black matte cloth is used as cowling on the robot, and as curtaining to hide any background scene. Thus, the "world" as seen by the cameras contains only one or more white balls against a black background. The result of this simple "early vision" strategy is a pair of pixel addresses containing the centroid of the single illuminated region seen by each camera. For the remainder of this presentation we will denote by W_k the function that takes a white ball against a black background into a pair of thresholded image plane regions and then into a pair of first order moments at the k^{th} field,

$$v_k := W_k\left(c(Cp_k)\right).$$

We use $p_k := F^{-\tau_f}\left(w_k\right)$ as an "extra" state variable to denote the delayed image of the ball's state due to image acquisition and processing delays.

Sensor Control Computational resources in the juggling system preclude examining more than about 1200 pixels from any given video field (our digitization system delivers individual fields at a rate of 60Hz). Thus the system is forced to process subwindows from the images to assure completion of the image processing task before the arrival of a new field. Figure 3 depicts the trivial sensor control strategy used in this initial design, which functions by centering the window for a new field over the location of the centroid from the previous field. This strategy implicitly presumes that objects do not move (or at least they do not move far) between images.

Triangulation In the initial implementation we chose to perform "exact" algebraic triangulation as defined by direct computation of $c^{\dagger}$ from (6). The resulting *spatial* position measurements were then passed directly to a linear observer. The result of application of $c^{\dagger}$ to the centroid data may be written as

$$\hat{C}(p_k) := \bar{b}_k = c^\dagger \circ W_k \circ c(Cp_k), \tag{8}$$

to make explicit the role of the *data processing* module.

State Estimation Due to digitization and processing latency, the image measurements generated by the *data processing* section are results from images that are at least one field (16ms) old. It follows that we ought to construct an observer which operates on this delayed data,

$$\hat{p}_{k+1} = F^{\tau_f}(\hat{p}_k) - G(C\hat{p}_k - \bar{b}_k), \tag{9}$$

where the gain matrix, $G \in \mathbb{R}^{6\times 3}$, is chosen so that $A_{\tau_f} + GC$ is asymptotically stable — that is, if the true delayed data, Cp_k, were available then it would be guaranteed that $\hat{p}_k \to p_k$[1].

3.1.2 Drawbacks

As detailed above, it is not the ball's position, b_k, which is input to the observer, but the result of a series of computations applied to the delayed copies of the cameras' image planes, $\bar{b}_k$. Prior to attempting two-juggle experiments, we ignored this "detail" and happily ran with the open loop sensory management procedures used to obtain data (8) [17]. It soon became clear that these procedures could not be similarly transparent in the more demanding domain of the two-juggle task. The practical limitations of our robot arm necessitated considerable enhancements to the vision subsystem, and getting these management issues right became one of the chief sources of difficulty.

For reasons detailed in [18] the considerable torque generating capabilities of our Bühgler arm did not prove sufficient to permit easily tracked ball trajectories in the two-juggle setting. We were forced to juggle higher (longer flight times between impacts) and to bring the two balls much closer together in space (shorter distance between impacts) than had been originally planned. This necessitated adding two new corresponding features to the vision system. First, we required an ability to sense and recover from out of frame events (a ball passing out of the field of view due to the height of the juggle). Second, we required that the system handle regularly occurring ball occlusions (two balls appearing at or near the same location in an image).

Neither the *data processing* nor the *sensor control* module described above are equipped with mechanisms suitable for handling either of these events. In particular the *data processing* module is incapable of recognizing *occlusion* events, and will happily produce erroneous measurements in their presence, while the naive *sensor control* strategy will never be able to reacquire a ball which leaves the field of view unless it returns near enough to the point of departure. The particular approach to solving this problem, which we will present below is a natural extension of our basic design.

[1] In principle, one might choose an optimal set of gains, G^*, resulting from an infinite horizon quadratic cost functional, or an optimal sequence of gains, $\{G_i^*\}_{i=0}^{k}$, resulting from a k-stage horizon quadratic cost functional (probably a better choice in the present context), according to the standard Kalman filtering methodology. Of course, this presumes rather strong assumptions and a significant amount of à priori statistical information about the nature of disturbances in both the free flight model (1) as well as in the production of $\bar{b}$ from $\hat{d}$ via the moment generation process. To date we have obtained sufficiently good results with a common sense choice of gains G that recourse to optimal filtering seems more artificial than helpful.

3.2 An Integrated Solution

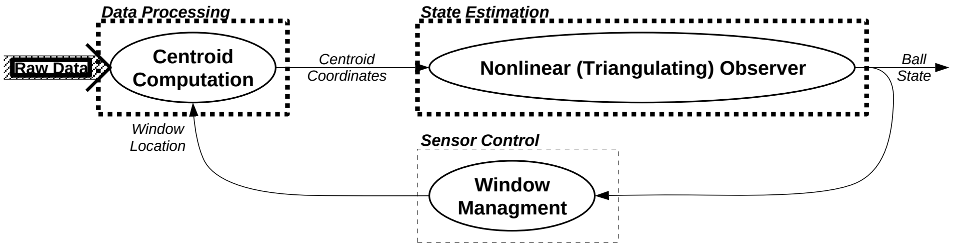


Figure 4: Working implementation: the active visual estimator.

This section presents the intuitively conceived modifications we made to the original sensing system described above. Individually, each of these represents a minor enhancement to the original system. However, getting them all to work in concert requires a greater amount of thought.

3.2.1 Data Processing: Making Use of Dynamic Segmentation

As mentioned above the fundamental weakness of the initial design centers around the inability of the *data processing* subsystem to recognize uninterpretable images. The necessary changes to "fix" this problem are not difficult and are presented here. However these modifications cause a fundamental change in the sensing operation and result in a ripple of changes throughout the remainder of the sensor system.

Occlusion and Out of Frame Events The construction of a system capable of handling *occlusion* and *out of frame* events must include the capability to either detect and reject images containing such events, or to reliably extract the relevant information in spite of these events. Clearly, in the case of out of frame events, there is no choice but to predict future behavior without new information, however our interest in exploring robust and extensible algorithms makes us disinclined to pursue more complicated recovery schemes even when they are possible. Having already committed to measuring the first order moments of a binary image as the primary method of localization, it is natural to extend this notion and use the zeroth and second order moments as simple and robust *occlusion detectors.* Under reasonably well-structured lighting conditions, the "ballness" of an image is easily determined by placing thresholds around the zeroth order moment and the the ratio of the eigenvalues of the matrix of the second order moments in conjunction with a test on the planar orientation of its eigenvectors.

The claim that the sensing system can be made more functional by making use of prediction both when data is unavailable (when a ball is not in sight) and when the data is difficult to interpret (when two balls pass close together in an image) is based on our belief that a unified approach to these problems will promote behavioral consistency and ultimately afford an analytical understanding. Additionally while a two-ball occlusion can be relatively easily disambiguated from geometric information, more balls or more complicated shapes will give rise to increasingly difficult and computationally intensive problems whose real-time solution will not be practical. Thus, we prefer to make a very coarse (and presumably, more robust) decision concerning when an occlusion has occurred, entrusting the dynamical model (the observer of Section 3.1.1) to provide sufficient information about where both balls will be in the future, thereby ensuring their reacquisition. As will be seen directly, this decision has consequences that set us on the path of building a "dynamical sensor."

This modification change amounts to augmenting the *data processing* segment of the system with a "confidence measure" suitable for deciding when the system has seen a ball and when it has not. Clearly the choice of "confidence" measures chosen here is simplistic and merely adequate to the task at hand. More sophisticated approaches have been proposed in the literature [2, 16]. The important idea to note is that when the sensing system controls how measurements are acquired, it must be capable of determining if a measurement was successful.

Active Segmentation In choosing to reject uninterpretable images at this low level we have implicitly assumed that the higher levels of the sensing system will be able to guide our future measurement efforts. Specifically we expect the *sensor control* module to supply clues sufficient to guarantee that a temporarily ignored object will be reacquired. In addition to providing a means to negotiate past losses of data, the availability of these clues makes the image processing task less complex by providing accurate initial estimates for an object's location. This "initialization" of the *data processing* subsystem allows for a more rational and localized search of the available data when attempting to extract meaningful features.

3.2.2 Sensor Control: Feedback for Active Vision

The idea of centering an image, whether in a sub-window or by moving a camera, is referred to as the "visual tracking" problem [6, 14]. As discussed above, the active control of the sensor is motivated by the need to provide suitable clues to the *data processing* subsystem so as to assure continued acquisition of useful measurements. What follows below are our intuitive ideas about the design and implementation of such a system. The more formal question of the stability of the ensuing (nonlinear) feedback system is the subject of Section 4,

Window Placement The guarantee of regular *occlusion events* (because the balls are purposefully juggled high and close together), coupled with the policy outlined above of ignoring data from occluded windows severely compromises the effectiveness of the simple previously acceptable window placement scheme. An obvious improvement results from using the estimates of the observer to place the windows. Namely, the windows in the next image to be processed are centered at a point formed by projecting the present state of the observer onto the camera image planes. Thus, the window location is now fedback from the output of the estimator whose inputs it provides. This connection of the observer back to the low-level data processing is exactly the *sensor control* module discussed above, and forms the "active vision" aspect of this system.

Window Size Adjustment Our inability to compute with more than a small percentage of the available pixels during the 16 msec interval between successive camera fields forces a tradeoff between the accuracy of the centroid data input to the observer and the possibility of an unnecessary and unrecoverable out-of-window event. This tradeoff is governed by the choice of sampling resolution or, equivalently, image plane window area. Intuitively, it seems clear that we ought to be able to develop some rational scheme for adjusting the sampling resolution in accord with an evolving set of error estimates. But what model of decision making offers an appropriate basis for such decisions, and where might one find a reasonable model by which to form the requisite estimates of error?

There are three principal sources of error in the sensing system. First, noise inevitably corrupts the image processing (e.g., distortions introduced by thresholding an imperfectly illuminated scene, or by insufficient spatial resolution). Second, the observer is itself compromised by parametric errors (e.g., the gravitational force, $\tilde{a}$ in (1) is obtained through our calibration procedure) and omissions (e.g., there is no model of spin during flight). Finally, these are exacerbated by the intermittent loss of input data that attends occlusion events (e.g., out-of-frame events may easily last in excess of 0.25 seconds).

Section 4 offers a formal presentation of the system theoretic ideas which support our current implementation. Fundamentally we grow the window area following any image plane measurement failure (i.e., an occlusion event), while the window area is shrunk following valid measurements. The exact size of the window needed to guarantee a successful future measurement is derived by bounding the current error in the *state estimator*, so as to ensure that the window will encompass the actual location of the ball. [2]

3.2.3 State Estimation: A Nonlinear (Triangulating) Observer

A central difference between the system presented in Section 3.1.1 and the one discussed here arises from the idea of discarding data from individual cameras whenever the image is "difficult" to interpret. The significant side-effect of this change is apparent when we look at the algebraic triangulator used to supply spatial ball positions to the linear observer. The system is unable to perform triangulation whenever date from *either* camera has been rejected, and thus new inputs can not be provided to the observer. Since it it is unlikely for data from both cameras to be invalid simultaneously, the discarding of questionable data from one camera has apparently forced the system to needlessly discard valuable data from the other.

Motivated by this apparent misuse of the available data we began investigating the use of partial data during an occlusion event. What resulted is a "triangulating observer" or "dynamical triangulator". Essentially this amounts to a nonlinear dynamical filter which is capable of making use of input from any number of cameras to update a state estimate for an observed linear dynamical system. The details of the development of this filter are the topic of Section 5. Needless to say such a filter can easily be given "zero-error" measurements whenever data is unavailable from a camera and continues to make use of all the available data. In addition to the

[2] Unfortunately, the larger the windows, the greater the chance of their overlapping and multiple balls being visible in a single window. The *data processing* system is augmented with an excision rule to removing intersecting regions from one window and assigning them exclusively to the other.

added capability of not being forced to ignore useful data, we are drawn to the notion of using a dynamical system to perform triangulation rather than computing an algebraic inverse. The potential for improved robustness through management of measurement uncertainty by replacing an algebraic operation with a dynamic filter has further motivated our interest in exploring this class of state estimators.

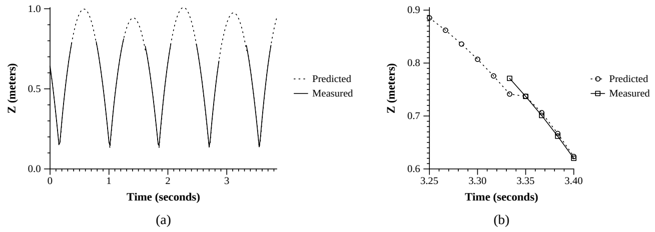


Figure 5: Measured and predicted (by the observer) ball heights for an out of frame juggling sequence (a), and an expanded view of a single recovery event (b).

3.3 Effect of the Modifications

We are convinced that the sensing enhancements discussed above have significantly contributed to our success at the two juggle task, as discussed in [17]. This Section documents the performance of the sensing system for the previously unmanageable situations discussed above.

Recovery from Out-of-Frame As mentioned above, the use of *state estimator* output to place the windows, has allowed the juggling height to be raised to the point that every juggle passes out of the field of view of our vision system. Figure 5 (a) and (b) depict exactly such a sequence. The top 0.25 to 0.4 seconds of each flight are outside the field of view, as is evident by the lack of position measurements during this period. Nevertheless the observer continues to predict the ball's location, and the ball is recovered as it passes back into the system's field of view. Figure 5(b) shows a detail of a single recovery. Evidently there is indeed a slight build up of prediction error (approximately 5 cm vertical error) over the near 0.5 second that this ball was outside of view. However since the measurement window has grown, this magnitude of error is readily accommodated.

Recovery from Ball-Ball Occlusions Similarly we have been been able to observe the occlusion events discussed above. Figure 6 and 7 depict the image plane tracks generated during an occlusion event. The small squares represent measurements assigned to ball 0, while the triangles are those associated with ball 1. The solid and dotted boxes are the windows used for moment calculations for ball 0 and 1 respectively. These are numbered corresponding to the temporal sequence of fields read. Figure 7 is a blow-up of a subregion of the right image plane shown in the previous figure, and is included so that the occlusion event (which occurs in the left camera) can be more clearly seen. In this particular sequence ball 0 (the squares) is rising

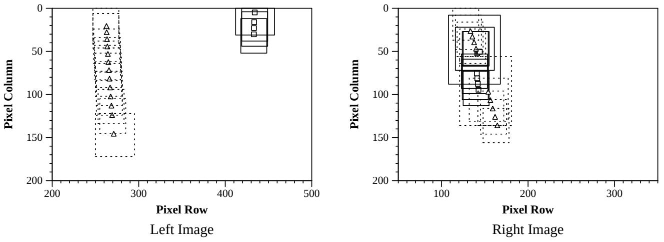


Figure 6: Left and Right image-plane tracks of a ball-ball occlusion event.

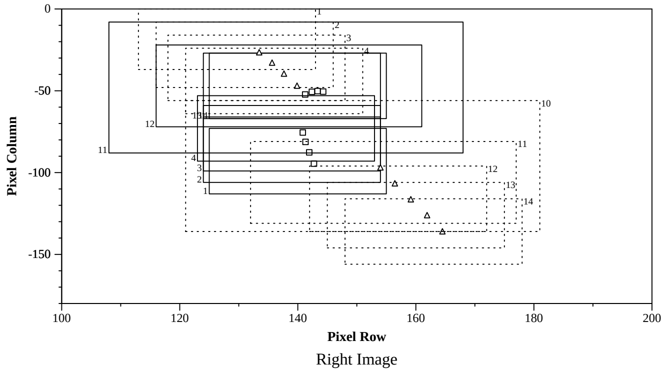


Figure 7: Expanded view of the left image-plane tracks showing the occlusion event.

towards its apex as ball 1 falls "behind" it causing an occlusion in the 5^{th} frame. [3] The balls remain occluded (lying within the overlap region between the two large windows) until the 10^{th} frame at which point ball 1 reappears from behind the search window for ball 0, and frame 11 when ball 0 becomes visible due to the search window for ball 1 shrinking and exposing it.

The Triangulating Observer The inclusion of the triangulating observer, which is discussed in great depth in Section 5 has afforded more reliable recovery from out of frame events. This is due to the ability of this observer scheme to make use of data from one camera even if there is no data from the other camera. Figure 8 demonstrates the difference between this observer and and the triangulator/linear-observer system in just such a situation. Figure 8(a) shows the overall flight of the ball as estimated by both observers, and measured by the triangulator (absence of the solid line indicates that the ball was out of frame). In this example the ball travels out of frame for approximately 0.2 sec. As can be seen in Figure 8(b) (a blowup of the ball returning into the field of view) the dynamical triangulator is capable of updating its estimate while the triangulator/observer pair are forced to simply predict the trajectory (note the differing behavior from 1.05 to 1.10 seconds). Significant reduction in tracking error then results as the ball reappears in both camera's fields of view at 1.10 seconds. This anecdotal picture is confirmed by experimental statistics. Figure 9 shows the mean and standard deviation of the norm squared tracking errors (position only) for the first four frames after recovery from an out

[3] To enhance visual clarity we have chosen to not show the windows that failed one of the "valid data" (i.e., zeroth or second order moment computation) tests and thus result in no input to the observer. Consequently, the windows "jump" from 4 to 11 and 4 to 10 for ball 0 and 1 respectively.

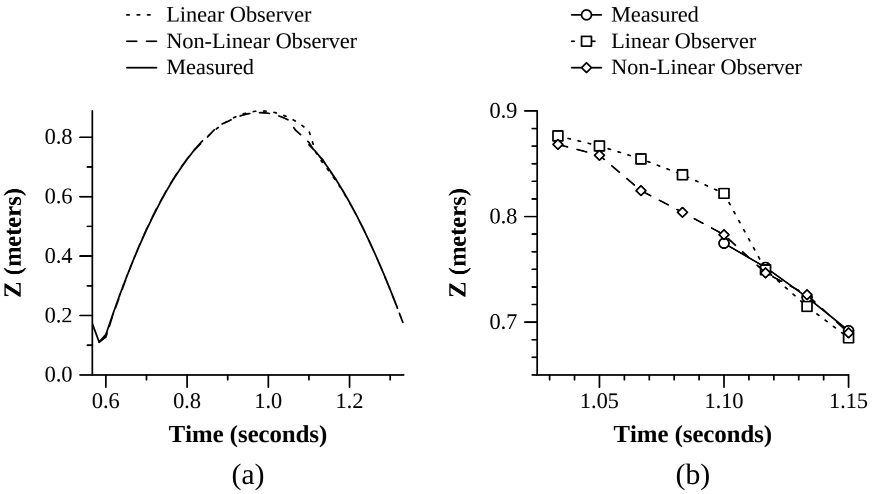


Figure 8: Experimental Data: Triangulated ball height and estimated ball height from both observers during recovery from a typical out of frame event.

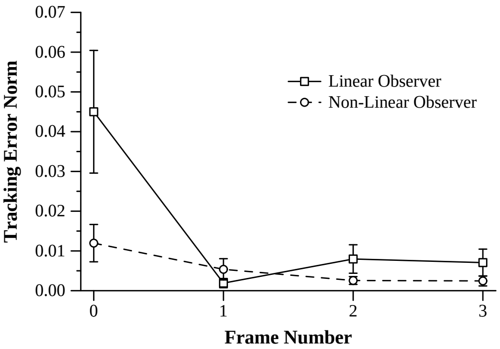


Figure 9: Experimental Data: Mean and standard deviation for the spatial observer errors immediately after recovery from out of frame, averaged over 102 events.

of frame event for 102 typical events.

4 “Active Vision:” Controlling the State of Attention

Whether choosing what segment of an image to process, where to look with a camera, or what camera to look with, many modern vision based systems incorporate an implicit control system – the control of the “state of attention” of the machine. In the case of our robot juggler this problem appears quite explicitly as a result of the limited real-time vision hardware. The machine is limited to only processing a small fraction of the total available data and must thus choose what data to process. This problem of *active vision* introduces a novel aspect of control to the sensing problem: the system responsible for control of attention must balance the benefit of examining only a small amount of sensor data against the risk of failing to generate useful measurements.

What follows is a detailed examination of this problem under the presumption that there are three noise sources (sensor noise, modeling inaccuracy, and measurement inaccuracy due to the "area of attention") to be balanced against the need for the state estimate to converge.

4.1 The Sensor Control Variables as a "State of Attention"

The *sensor control* module of Figure 2 is responsible for controlling the locus and extent of the image plane windows used for information extraction by the *data processing* subsystem. Thus, we tentatively define a window's *state of attention* at some field interval, k, as the pair

$$a_k = \left(\hat{b}_k, \rho_k\right) \in I\!R^3 \times I\!R^+ \tag{10}$$

where $\hat{b}_k$ denotes an estimate of where the falling ball is expected to appear, and the positive scalar ρ_k is a measure of "certainty" of this estimate. With respect to a norm, $\|\cdot\|_M$, which will be defined below, a_k induces two windows on the two camera image planes including all stereo image pixel pairs, x, in the set

$$\mathcal{N}(a_k) := \left\{x \in c(I\!R^3) : \|\hat{b}_k - c^{\dagger}(x)\|_M \leq \rho_k\right\}.$$

The *data processing* subsystem will process these windows, and if the "ballness" tests (zeroth and second order moment tests) are passed, the first order moments will be passed to the *state estimation* to be interpreted as a spatial position. Otherwise, an "empty window" will be reported. For the sake of notational simplicity, we will denote the situation that first order moments are successfully formed inside the windows of the k^{th} camera field as

$$c(b_k) \in \mathcal{N}(a_{k-1}).$$

The dependence of the k^{th} measurement on a_{k-1} immediately suggests the *dynamics* intrinsic to the general sensor management problem – which appears here as mere delay. Regardless of how it is computed, the state of attention, a_k must be assembled from information derived from existing sensory observations. Thus, the acquisition of new data is necessarily mediated by old knowledge and a feedback loop is formed.

For a suitable norm, we look back to the stabilized observer equations (9). Because the poles of the closed loop observer have been placed within the unit circle there exists a positive definite symmetric matrix, M, such that

$$\left[A_{\tau_f} + GC\right]^T M \left[A_{\tau_f} + GC\right] < M,$$

and we will denote the Euclidean norms induced by this matrix as

$$\|p\|_M \triangleq \left(p^T M p\right)^{1/2}; \qquad \|A\|_M \triangleq \sup_{\|p\|_M = 1} \|Ap\|_M$$

For ease of exposition we introduce the notational conventions,

$$\alpha \triangleq \|A_{\tau_f}\|_M; \qquad \bar{\alpha} \triangleq \|A_{\tau_f} + GC\|_M \tag{11}$$

and assume, purely for further notational convenience, that the poles of the closed loop observer equation (9) have been placed on the real line with multiplicity two with the consequence that

$$\|\left[A_{\tau_f} + GC\right]^{-1}\|_M = 1/\bar{\alpha}.$$

4.2 Observer Errors from a Noisy Model

The task at hand is to develop a control scheme for updating the state of attention, a_k as a function of its previous value and presently available data. To do so we must append to our previous state estimation procedure some notion of its changing degree of certainty. Thus, reconsider the Newtonian flight model (1), with the addition of both a process and a sensor noise model. We wish to model the inaccuracies in the Newtonian flight law as well as the salient features of the inaccuracies in ball position measurement introduced through the use of the camera. The latter includes two central phenomena: the absence of data when the ball lies outside of its assigned window; and the imprecision of spatial localization as the size of the window grows (and either delay grows or resolution shrinks correspondingly). For present exploratory purposes, we will be content with a crude deterministic representation of the imprecision inherent in these process and sensor models.

We substitute for (1) and (8) the system

$$\begin{array}{ll} w[(j+1)\tau_r] & = F^{\tau_r}\left(w(j\tau_r)\right) + n_N(j\tau_r) \\ p_{k+1} & = w[k\tau_f] \\ \bar{b}_k & = \hat{C}_k\left[p_k + n_S(\rho_{k-1})\right]. \end{array} \qquad (12)$$

As a first crude model for the failings of the putative Newtonian free-flight model (1) we take n_N to be a bounded deterministic sequence of uncontrolled inputs (perhaps generated via a map on the state space), and n_S to be the sensor noise introduced by thresholding a finite resolution image before computation of the moments. Because the window resolution must decrease as the window size increases (as a consequence of subsampling), n_S is non-decreasing in its argument. Since no subsampling is required for sufficiently small windows, n_S is a positive constant for small values of its argument. These considerations suggest an affine model of sensor noise as a function of window radius.

$$\|n_S(\rho_k)\|_M \leq \nu_0 + \nu_1\rho_k. \qquad (13)$$

We choose to ignore the details of how $c(\cdot)$ and $c^{\dagger}(\cdot)$ influence the creation of errors in the measurement of $\bar{b}_k$, since this would require a careful assessment of the reflectance properties of the balls – a distant second order effect given the current structured lighting. In contrast we are greatly concerned with developing correct window management logic, and we will explicitly embed the influence of $W(\cdot)$ in $\hat{C}$ as follows.

The deterministic output map, $\hat{C}_k$ returns the value $C = [I, 0]$ as in (1) when the body's image is in the examined area of the image plane, and vanishes otherwise:

$$\hat{C}_k \triangleq \left\{ \begin{array}{ll} C & : c(b_k) \in \mathcal{N}(a_{k-1}) \\ 0 & : c(b_k) \notin \mathcal{N}(a_{k-1}) \end{array} \right. . \qquad (14)$$

This models the salient behavioral features of the *data processing* subsystem introduced in Section 3.1.1, as it returns no data (zero) when an "out of frame" event occurs. This results in the observer simply extrapolating the present state estimate in such situations. The resulting observer takes the same form as (9) only with $\hat{C}_k$ from (14) incorporated,

$$\begin{array}{ll} \hat{p}_{k+1} & = F^{\tau_f}(\hat{p}_k) + G(\bar{b}_k - \hat{C}_k \hat{p}_k) \\ \hat{w}(k\tau_f + j\tau_r) & = F^{\tau_f + \iota_k + j\tau_r}(\hat{p}_k), \\ & \qquad j = 0, 1, ..., \tau_f + \iota_{k+1} - \iota_k \\ \hat{b}_k & = CF^{\tau_f}(\hat{p}_k). \end{array} \tag{15}$$

Here, we distinguish between the state estimate, $\hat{w}(\cdot)$, that is sent forward to the juggling algorithm, and the attention variable, $\hat{b}$, that will be sent back to the *sensor control* module. The robot gets $\hat{w}(k\tau_f)$ as soon as it is formed, with future predictions being made at the faster physical rate, τ_r. The *sensor control* module will make use of $\hat{p}_k$ in the form of $\hat{b}_k$ to handle the $(k+1)^{st}$ image.

The result is a system with two distinct kinds of error, [4] each with its own causes and effects. The first is the standard error due to the observer,

$$\tilde{p}_k \triangleq p_k - \hat{p}_k,$$

and is governed by the dynamics

$$\begin{array}{ll} \tilde{p}_{k+1} & = \left(A_{\tau_r} + G\hat{C}_k\right) \tilde{p}_k + n_k \\ & \quad n_k \triangleq Gn_S(\rho_{k-1}) + n_N[(k-1)\tau_f]. \end{array} \tag{16}$$

Denoting the present error magnitude by $\vartheta_k \triangleq \|\tilde{p}_k\|_M$, we can conclude that

$$\begin{array}{ll} \vartheta_{k+1} & \leq \lambda_k \vartheta_k + \|n_k\|_M \\ & \qquad \lambda_k \triangleq \left\{ \begin{array}{ll} \bar{\alpha} < 1 & : c(b_k) \in \mathcal{N}(a_{k-1}) \\ \alpha > 1 & : c(b_k) \notin \mathcal{N}(a_{k-1}) \end{array} \right. , \end{array} \tag{17}$$

(α and $\bar{\alpha}$ are defined in (11)) and it follows that the necessary and sufficient condition on $\hat{C}_k$ and λ_k for a measurement to be successfully taken may now be expressed as

$$c(b_k) \in \mathcal{N}(a_{k-1}) \Longleftrightarrow \|C^T(Cw[(k-1)\tau_f] - \hat{b}_{k-1})\|_M < \rho_{k-1}. \tag{18}$$

Thus, there is a second sort of error associated with this event. It is due to the conjunction of process noise with time delay in the formation of the extrapolated state estimate. For, assuming $\|n_N\|_M$ is bounded above by the scalar ν_N, we have

$$\begin{array}{ll} \|C^T(Cw[(k-1)\tau_f] - \hat{b}_{k-1})\|_M & \leq \|w[(k-1)\tau_f] - F^{\tau_f}(\hat{p}_{k-1})\|_M \\ & \leq \alpha\left(\vartheta_{k-1} + \tau_f \nu_N\right). \end{array} \tag{19}$$

It follows that if ρ_{k-1} is at least as large as the last expression, we are guaranteed (within the limits of our noise model) that the k^{th} window will not be empty — that condition (18) will hold.

[4] Note that there is actually a third sort of error, which concerns the quality of the estimate passed forward to the robot. If $\tilde{w}_k \triangleq w(k\tau_f + \iota_k) - \hat{w}(k\tau_f)$ we have, $\|\tilde{w}_k\|_M \leq \alpha^{\iota_k}(\vartheta_k + (\tau_f + \iota_k)\nu_N)$, where $\tau_r \leq \iota_k \leq \tau_f$. Thus, $\|\tilde{w}_k\|_M$ is a non-decreasing function of both ϑ and ρ. But this error is never seen in the sensory loop.

4.3 Window Radius Control

The construction of a functional observer of the form presented in (12) necessitates the implementation of a *sensor controller*. Specifically, this amounts to choosing window sizes, ρ_k, and locations, $\hat{b}_k$, in such a fashion that the acquisition of new measurements can be guaranteed in conjunction with the estimated state converging to the actual state.

4.3.1 Certainty Estimates from a Parallel Observer

The result of (19) implies that ρ_k should be set in relation to ϑ_k in order to insure data to the observer. But, unfortunately, we are not in possession of the error magnitude, ϑ, for the very reason that we were led to build an observer in the first place (our inability to measure ball velocities). Since $\hat{p}$ represents our only knowledge of p, the best estimate of ϑ is 0 as matters stand presently. To address this deficit, we will build a second state estimator and attempt to construct and estimate of ϑ by comparing the two.

Using the invertibility of the observability matrix,

$$\Theta := \begin{bmatrix} C \\ CA_{\tau_f} \end{bmatrix},$$

we may define a very different estimate of p taking the form

$$d_k = F^{\tau_f}\left(\Theta^{-1}\left(\begin{bmatrix} \bar{b}_{k-1} \\ \bar{b}_k \end{bmatrix} - \begin{bmatrix} 0 \\ Ca_{\tau_f} \end{bmatrix}\right)\right).$$

This is a dead-beat observer in the sense that $\tilde{d}_k \triangleq p_k - d_k$ converges to zero in two steps from all initial estimates, d_0 in the absence of noise, $n_S = n_N = 0$.

Through careful comparison of the estimates provided by these two observers (as detailed in Appendix B) we are led to define a worst case estimate for ϑ as

$$\hat{\vartheta}_{k-1} \triangleq \left[\|\hat{p}_k - d_k\|_M + \nu_\Delta(\rho_{k-1}, \rho_{k-2})\right]/\bar{\alpha}, \tag{20}$$

where

$$\nu_\Delta(\rho_{k-1}, \rho_{k-2}) := \|n_{k-1}\|_M + \alpha\tau_f\nu_N + \frac{\alpha}{\|\Theta\|_M}\left(\nu_N + \|n_S(\rho_{k-1})\|_M + \|n_S(\rho_{k-2})\|_M\right),$$

which guarantees that $\hat{\vartheta}_{k-1} \geq \vartheta_{k-1}$.

4.3.2 Control of Window Radius

Equipped with a worst case estimate for ϑ, we are now in a position to adjust ρ. According to the previous calculations (19), a window radius management strategy that achieves the relation

$$\rho_k \geq \alpha\left(\vartheta_k + \tau_f\nu_N\right)$$

guarantees data to the observer at step $k+1$. Noting that ϑ_k is causally determined by ρ_k, and thus cannot be estimated directly by the procedure (20) at stage k, we appeal to (17) and note that the desired relation is implied by

$$\rho_k \geq \alpha \left(\lambda_{k-1}\vartheta_{k-1} + \|n_{k-1}\|_M + \tau_f \nu_N\right).$$

This demonstrates that the radius adjustment procedure

$$\rho_k = \alpha \left(\lambda_{k-1}\hat{\vartheta}_{k-1} + \|n_{k-1}\|_M + \tau_f \nu_N\right) \tag{21}$$

will always yield a window large enough to capture the next centroid, up to the limits of the error models employed.

4.3.3 Boundedness of Estimator Errors

This then leaves the question of observer convergence. Recall that as ρ increases, the quality of the robot estimates deteriorates. Eventually, the recourse to subsampling might begin to have a net destabilizing effect through the injection of noise represented by n_k in (17). We must show that the coupled dynamical system (17), (21) remains stable.

As derived in Appendix B.2 the coupled dynamics for ϑ_k and ρ_k may be bounded by

$$\begin{aligned} \vartheta_{k+1} &\leq \lambda_k \vartheta_k + \nu_1 \rho_{k-1} + \gamma \nu_0 + \nu_N \\ \rho_{k+1} &\leq \alpha \left(\tau_f \nu_N + \gamma(\nu_0 + \nu_1 \rho_{k-1}) + \nu_N + \tfrac{\lambda_k^2}{\bar{\alpha}}\left[\vartheta_k + 2\nu_\Delta(\rho_k, \rho_{k-1})\right].\right) \end{aligned}$$

Moving to the coordinate system, $x \triangleq [\chi_1, \chi_2, \chi_3]^T$, where $\chi_1(k) \geq \vartheta_k$ bounds the actual Lyapunov magnitude of (15) and $\chi_2(k) \geq \rho_k$, $\chi_3(k) \geq \rho_{k-1}$ represent bounds on the most recent window radius values, we obtain the dynamics

$$\begin{aligned} x(k+1) &= Q_k x(k) + r \\ Q_k &\triangleq \begin{bmatrix} \lambda_k & 0 & \nu_1 \\ \frac{\alpha\lambda_k^2}{\bar{\alpha}} & \alpha\nu_1 g_1 & \alpha\nu_1 g_2 \\ 0 & 1 & 0 \end{bmatrix} \\ r &\triangleq \begin{bmatrix} r_1 \\ r_2 \\ 0 \end{bmatrix}, \end{aligned} \tag{22}$$

where the symbols $g_i, r_i, i = 1,2$ denote constants derived from the computations developed above.

By construction of the radius adjustment procedure (21), the state of this system enters a region where $\lambda_k = \bar{\alpha} < 1$ after an initial transient. Now, elementary root locus analysis of the characteristic polynomial of this system,

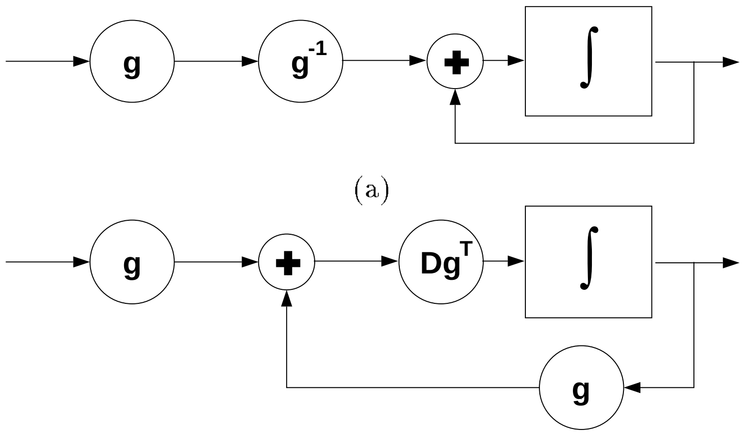
$$s^2(-\bar{\alpha} + s) + \alpha\nu_1\left[(g_2 - 1)\bar{\alpha} + (\bar{\alpha}g_1 - g_2)s + g_1 s^2\right]$$

shows that the matrix Q has roots in the unit circle of the complex plane for small enough values of ν_1 (they originate at $\{\bar{\alpha}, 0, 0\}$). This implies that if the noise coefficient, ν_1 is sufficiently small relative to the other parameters then the window management system succeeds in keeping the windows large enough to retain the required image, but not so large as to destabilize the estimation procedure.

4.4 Implementation

As reported in Section 3.3, we have implemented and performed experiments on a system similar to that described here. The particular implementation was constructed as a precursor to the analysis presented here. As such the experimental implementation uses a slightly less precise method to adjust the window sizes than as presented in (21). In particular the experimental implementation does not make explicit use of a parallel observer to generate confidence estimates, but rather makes use of a small finite state automaton to control window size based on a less carefully designed model of the recent history of measurement success and failure. The implementation is however structurally equivalent, and has provided significantly enhanced capability in the sensing system as documented in Section 3.3. It is worth noting that prior to implementing a window management strategy of this type our machine had been incapable of tracking two falling balls for a sufficient period of time to allow for experimental verification of the underlying juggling algorithm.

5 Dynamic Triangulation



(a)

(b)

Figure 10: Two approaches to inverting an information preserving nonlinear function: (a) direct or algebraic inversion; (b) dynamic inversion.

We endeavor here to embed the triangulation process directly in an observer and thereby making use of all the available data at all times, while continuing to guarantee convergence of the state estimates. As discussed at the end of Section 3 the previously mentioned “waste” of data arose from the use of an “algebraic” inverse to transform image plane measurements into spatial positions. The alternative we present here is based on performing this inversion implicitly in a dynamical filter. [5]. Figure 10 demonstrates the underlying structural difference between these approaches.

Section 4 formalized the notion of control of attention for an observer that receives data (a measurement of the spatial location of the ball) or not, depending only on the placement of windows. This is an overly simplified processing architecture (detailed in Figure 11) appropriate to the use of $c^\dagger$ in converting image plane data to spatial ball locations: after all $c^\dagger$ cannot be computed if data is not available from both cameras. More realistically, an occlusion or out of frame event is likely to inhibit data from one camera or the other, rather than both. Thus, there is apparent benefit to be gained from constructing an observer which can make use of this partial data when it is available. What follows is an approach for achieving this greater efficiency.

Underlying the new estimation technique is the simple idea of augmenting the standard (linear) Newtonian flight model, $\ddot{b} = \tilde{a}$, with a nonlinear "output map," $v = c(b)$, and constructing a non-linear observer which updates its state estimates based on the image plane data rather than a spatial measurement derived through triangulation. The structure for this "new" observer is shown in Figure 11. The significant change here is to abandon the use of an algebraic triangulation function to invert $c(b)$ and instead revert to using a dynamical system to smooth, predict, and perform this inversion, all through its update law for the estimated state. The expectation here is that beyond the efficiency achieved by not "wasting" good data, such a system will exhibit better noise immunity since it does not directly attempt to invert c.

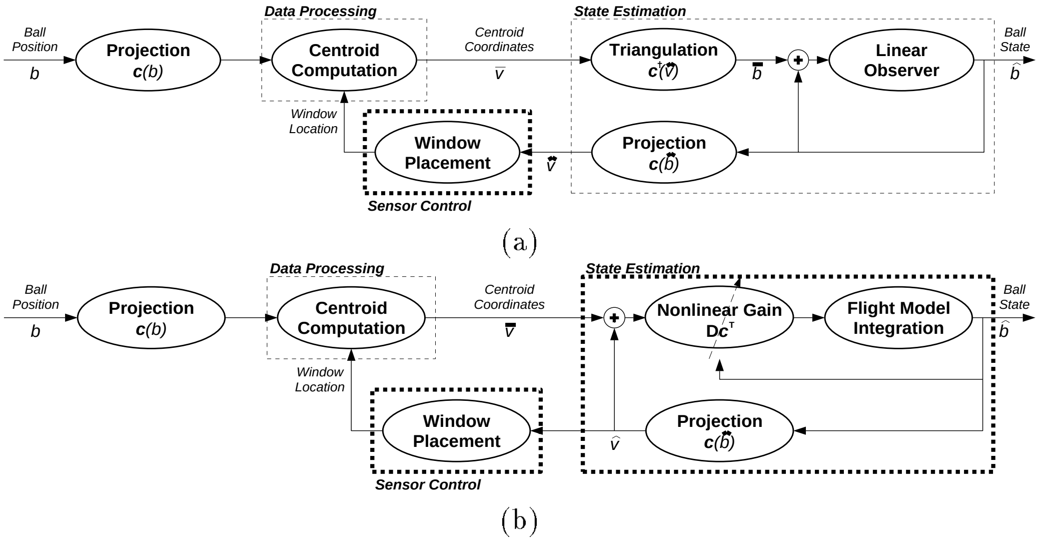


Figure 11: Structure of the observer scheme from Section 4 (a), and Structure of the triangulating observer (b).

How should we think about this scheme fitting in the class of visual servoing systems? Let us agree to define a visual serving task as a problem wherein some measured visual data is to be aligned with some desired visual data value. Then the system we are considering clearly fits that

[5] In much the same way an integrator in a feedback loop can be used to implicitly form the derivative of an input signal, or an analog computer can continuously find root of a polynomial

definition: "move" the estimate of a ball's position to visually align with the measured data. Traditionally [11, 15, 9, 10] researchers working in this are have had static goals for environments they can manipulate (i.e. move the robot until the the scene looks like "this"). Our problem is the dual of this in much the same way a linear observer is the dual of a linear controller – we must manipulate our state estimate so that it coincides with the visually acquired data. This section offers a detailed look at the mathematics behind this class of problems, followed by some simple example systems to which existing theory may be applied, and concludes with an algorithm relevant to the physical problem at hand, whose success has not yet been theoretically explained.

5.1 A Property of the Perspective Projection

Recall from (7) that the stereo camera transformation, c, is formed by stacking together the perspective projections due to the two individual cameras. In this section we note that

$$\begin{aligned} c(\hat{b}) - c(b) &= \Lambda(\hat{b}, b) C(\hat{b}) \left(\hat{b} - b\right) \\ &= \Lambda(b, \hat{b}) C(b) \left(\hat{b} - b\right), \end{aligned} \tag{23}$$

where $C(b)$ is the jacobian of c evaluated at b and

$$\Lambda(\hat{b}, b) \triangleq \begin{bmatrix} \frac{\Pi_3 \, {}^1H_0 \, \hat{b}}{\Pi_3 \, {}^1H_0 \, b} I_2 & 0 \\ 0 & \frac{\Pi_3 \, {}^2H_0 \, \hat{b}}{\Pi_3 \, {}^2H_0 \, b} I_2 \end{bmatrix}.$$

This fact emerges directly from computation. Given b lying in the frame of reference of a camera with focal length f, we have

$$\pi_f(b) \triangleq \frac{f}{b_3} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}. \tag{24}$$

The Jacobian of this projection is then given by

$$D_b \pi \left(b\right) = \frac{f}{b_3} \begin{bmatrix} 1 & 0 & -\frac{b_1}{b_3} \\ 0 & 1 & -\frac{b_2}{b_3} \end{bmatrix}. \tag{25}$$

Expanding the right hand side of the top row of (23) in these coordinates gives

$$f \frac{\hat{b}_3}{b_3} \left(\frac{\hat{b}_1}{\hat{b}_3} - \frac{b_1}{b_3} \right), \tag{26}$$

and similar results follow for the remainder of the rows. This establishes the original assertion.

Throughout the remainder of this section we will assume that Λ is positive definite. Geometrically, this implies that both b and $\hat{b}$ always lie on the same side (front/back) of all the cameras at all times. In practice this is not an unrealistic assumption: it merely requires that neither the actual object cross the singularity in (7) nor that the initial errors in the observer system become so large as to cause the estimated object location to cross this same singularity. Formally, this assumption allows us to assert that

$$(\hat{b} - b) C^T(\hat{b})(\hat{v} - v) > 0. \tag{27}$$

5.2 A First Order Observer

Consider the dynamical system described by

$$\begin{array}{rcl} \dot{b} & = & Ab + u \\ v & = & c(b). \end{array} \tag{28}$$

This system is "fully-measurable" – that is, algebraic triangulation can fully reconstruct it's state b. But it seems helpful to underscore the utility of (23) by considering an observer for this system that filters the resultant state estimates and allows for estimation during hypothetical partial loss of measurements. In fact this class of system has formed the central focus for a majority of the current visual servoing research [1, 12]. An observer for this class of system takes the form

$$\begin{array}{rcl} \dot{\hat{b}} & = & A\hat{b} + u - KC^T(\hat{b})\,(\hat{v} - v) \\ \hat{v} & = & c(\hat{b}), \end{array} \tag{29}$$

with $K \in I\!R^{3\times 4}$. Forming the error dynamics for $\tilde{v} \stackrel{\triangle}{=} \hat{v} - v$ we have

$$\dot{\tilde{b}} = A\tilde{v} - KC^T(\hat{b})\,(\hat{v} - v)\,. \tag{30}$$

Making use of (23) allows us to substitute for $(\hat{v} - v)$, and results in

$$\dot{\tilde{b}} = \left(A - KC^T(\hat{b})\Lambda(\hat{b}, b)C(\hat{b})\right)\tilde{b}. \tag{31}$$

From (27) we know $C^T(\hat{b})\Lambda(\hat{b}, b)C(\hat{b})$ is positive definite from which it follows that there exists a K such that[6] $\lim_{t\to\infty} \tilde{b} = 0$.

5.3 An Observer for Mechanical Systems with Linear Dynamics

In contrast to the completely measurable system presented above, let us now reconsider the system, $\ddot{b} = \tilde{a}$, written more generally as

$$\begin{array}{rcl} \dot{b}_1 & = & b_2 \\ \dot{b}_2 & = & A_1 b_1 + A_2 b_2 + u \\ v & = & c(b_1), \end{array} \tag{32}$$

where b_1 and b_2 represent the position and velocity of the object respectively. This system is of particular interest since it includes our model for the ball falling under the influence of gravity. The associated observer now takes the form

$$\begin{array}{rcl} \dot{\hat{b}}_1 & = & \hat{b}_2 - \Gamma_1 C^T(\hat{b}_1)\,(\hat{v} - v) \\ \dot{\hat{b}}_2 & = & A_1\hat{b}_1 + A_2\hat{b}_2 + u - \Gamma_2 C^T(\hat{b}_1)\,(\hat{v} - v) \\ \hat{v} & = & c(\hat{b}_1), \end{array} \tag{33}$$

[6] Note, due to the time-varying nature of (31), the choice of stabilizing K is necessarily influenced by the initial conditions. In practice a reasonable bound could be placed on the initial errors such that a suitably large fixed K may readily be chosen.

with gain matrices Γ_1 and Γ_2 free to be chosen. Proceeding as above we take differences to determine the error dynamics

$$\begin{aligned} \dot{\tilde{b}}_1 &= \tilde{b}_1 - \Gamma_1 C^T(\hat{b}_1)\,(\hat{v} - v) \\ \dot{\tilde{b}}_2 &= A_1\tilde{b}_1 + A_2\tilde{b}_2 - \Gamma_2 C^T(\hat{b}_1)\,(\hat{v} - v)\,, \end{aligned} \tag{34}$$

which simplifies to

$$\begin{aligned} \dot{\tilde{b}}_1 &= \left(I - \Gamma_1 C^T(\hat{b}_1)\Lambda(\hat{b}_1, b_1)C(\hat{b}_1)\right)\tilde{b}_1 \\ \dot{\tilde{b}}_2 &= \left(A_1 - \Gamma_2 C^T(\hat{b}_1)\Lambda(\hat{b}_1, b_1)C(\hat{b}_1)\right)\tilde{b}_1 + A_2\tilde{b}_2. \end{aligned} \tag{35}$$

While the convergence properties of the “first order observer” (30) follow directly from (27), the steady state properties of (35) are more difficult to establish. Numerous simulation studies and physical experiments indicate that the system converges, and an analysis of the assumptions under which convergence can be proven is currently in progress.

5.4 Implementation

Although the analysis of the previous section is at best in its infancy we proceeded, following a number of promising simulations, to construct a functional implementation of the observer described above. As usual, the real world departs from the assumptions underlying these models in certain important regards. What follows is a brief discussion of the differences between the previous section and the actual system, along with both experimental and simulation results demonstrating the utility and pitfalls for this type of observer.

5.4.1 Choice of Observer Gains

Having no immediate insight at the outset concerning choice of the gain matrices Γ_1 and Γ_2 in (33), we chose to use the same gains as for the linear observers. Poor convergence in our first simulations demonstrated that this simple choice was inadequate. The primary cause for this effect was that the spatial dependence of $C(\hat{b})$ leads to widely differing effective gains depending on the ball's location in space. We were able to successfully compensate for this by making use of non-linear gain matrices of the form

$$\Gamma = \Gamma_0 \left(C^T(\hat{b})C(\hat{b})\right)^{-1}.$$

This essentially amounts to performing local triangulation (i.e. $\Gamma C^T(\hat{b})$ is the linear approximation to $c^\dagger$ at $\hat{b}$), and dramatically improved the convergence behavior of the observer.

5.4.2 Time Sampled Implementation for Second-Order Systems

Real cameras are not continuous time devices – the affordable devices we are interested in generally take snapshots of the world at a fixed sampling rate, in our case, 60 Hz. Since the

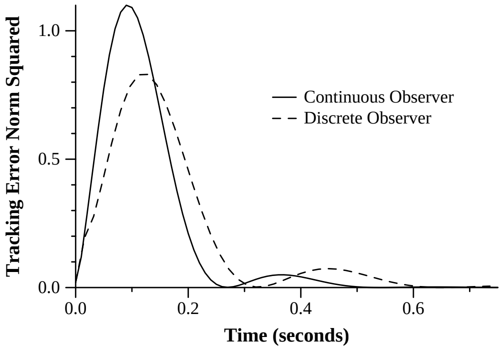
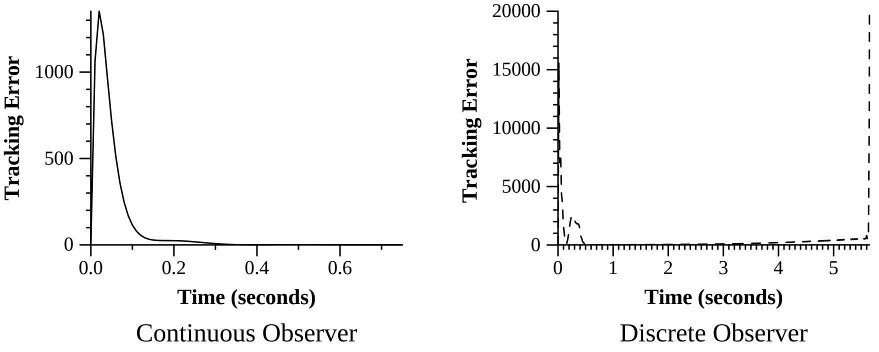


Figure 12: Simulation: Convergence of the continuous and discrete time observers for small initial error.

observed system's motion is significant relative to this rate (near impact, the ball often travels in excess of 10 cm between successive images), sampling considerations cannot be ignored. For the observer of Section 3.1.1 (with explicit triangulation) implementation with sampling presents no problem since the dynamical system we are observing is linear. Traditional discrete time systems theory affords a reliable observer (9). However no analogous theory is available for our new nonlinear dynamic triangulator, even were the theoretical questions of Section 5.3 entirely resolved.



Continuous Observer

Discrete Observer

Figure 13: Simulation: Convergence of the continuous and discrete time observers for larger initial error.

In the absence of any theory we have studied numerically both the continuous and discrete time systems. Figures 12 and 13 demonstrate how a change in the initial conditions can result in instability for the discrete system, while the continuous version remains well behaved. Figure 12 depicts a case where the discrete and continuous system demonstrate comparable behavior for identical gains and small initial errors, Figure 13 demonstrates that the same systems can display markedly different behavior for different initial conditions. In this particular example the continuous system converges reasonably quickly, while the discrete version initially behaves reasonably well, then slowly begins to fail until 5.5 seconds, when it "explodes".

5.4.3 Integration in the Juggling System

As pointed out in Section 3.3 the inclusion of this new type of *state estimation* system has lead to improvements in the behavior of the overall juggling system. The experimental validation of this approach has further motivated us to consider extensions of this idea to other visual sensing problems. One example which seems to naturally fit this model is the problem of integrating sparse data from a large number of sensors, or "sensor fusion" problem.

A Vision System Calibration

In the course of developing the sensing system discussed in this paper, we have been led to re-formulate a very attractive coordinated camera-arm calibration scheme originally proposed by Hollerbach [4]. At calibration time, one supposes that some point on the robot's gripper (that we will take to be the origin of the "tool" frame) is marked with a light reflecting material in such a fashion as to produce an unmistakable camera observation — a four vector, $c \in \mathbb{R}^4$ comprised of these two image plane measurements. The problem is to determine the kinematic parameters, $k \in \mathbb{R}^{8+3(m+1)}$, that characterize the robot chain as well as the relative camera frame relationship and camera focal lengths by comparing measured camera values with the joint space locations that produced them.

The Setting Denote by g_k, the forward kinematic transformation of the kinematic chain that expresses the robot's tip marking with respect to the base frame (that we take to be the frame of the "right" hand cameras with no loss of generality). According to the Denavit-Hartenburg convention, the parameter vector, $(k_1, ..., k_{m+1}) \in \mathbb{R}^{3(m+1)}$, that characterizes this function appears in the form

$$g_k(q) = \left(\prod_{i=1}^{m+1} H_i(\theta_i)\right) \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}; \qquad H_i(\theta_i) \triangleq \exp\left\{\theta_i \sum_{j=1}^{3} J_{ij} k_{ij}\right\},$$

where θ_i is a joint variable and J_{ij} is a constant 4×4 array whose exponent yields the homogeneous matrix representation of the unit screw scaled by parameter k_{ij}. If these $3(m+1)$ parameters were known then g_k would yield for every jointspace location, $q = (\theta_1, ..., \theta_n)^T \in \mathcal{Q}$, the homogeneous representation of the tool frame origin in base frame coordinates.

Now denote by H_0 the homogeneous matrix representation of the screw relating the "left" hand camera frame to the base frame,

$$H_0 = \exp\left\{\sum_{j=1}^{6} k_{0j} J_{0j}\right\},$$

where J_{0j} constitutes an arbitrary basis for the Lie Algebra corresponding to the group of rigid transformations and $\tilde{k}_0 \in \mathbb{R}^6$ parametrizes the relative camera frame transformation matrix accordingly. The camera transformation is now characterized by the parameters $k_0 = (k_{00}, k'_{00}, \tilde{k}_0) \in \mathbb{R}^8$ that appear in the the stereo projective transformation, $c : \mathbb{R}^3 \to \mathbb{R}^4$, that for a given camera pair associates with each spatial point a pair of ("left" and "right" camera) planar points. Specifically, let Π, π denote the projections from $\mathbb{R}^4$ that pick out, respectively, the first two, and the third coordinate, of a homogeneous representation of a point. The camera transformation may be written as

$$c(w) = \begin{bmatrix} \Pi(w)/k'_{00}\pi(w) \\ \Pi(H_0 w)/k_{00}\pi(H_0 w) \end{bmatrix}.$$

This function admits a family of pseudo-inverses $c^{\dagger} : I\!\!R^4 \rightarrow I\!\!R^3$, whose effect on the camera image plane, $c(I\!\!R^3) \subset I\!\!R^4$, returns the original spatial point — that is $c^{\dagger} \circ c$ is the identity transformation of $I\!\!R^3$ — and whose effect off the camera image plane is to return the "closest" spatial point to that four-vector with respect to a suitable metric.

A Modified Procedure Hollerbach's proposed procedure tested in simulation of a planar arm, [4], calls for recording some number of joint-space/camera-image pairs, $\mathcal{D} = \{(q_l, c_l)\}_{l=1}^{n}$, and then performing a Newton-like numerical descent algorithm on the cost function

$$\sum_{l=1}^{n} \|c^{\dagger}(c_l) - g_k(q_l)\|^2.$$

When we attempted to implement this procedure for the three degree of freedom Bühgler arm, we found that the procedure was extremely sensitive numerically.

Instead, we have had great success with a variant on this idea that substitutes a cost function in the stereo camera image space,

$$\sum_{l=1}^{n} \|c_l - c \circ g_k(q_l)\|^2,$$

for the previously defined workspace objective. We have been using this procedure on average several times a month (the experimental apparatus is frequently torn down and put back together again to incorporate new hardware, necessitating continual re-calibration) for the last six months with very good results. Starting from eyeball guesses of $k = (k_0, k_1, k_2, k_3, k_4)$, we have been able to achieve parameter estimates that give millimeter accuracy in workspace after a short period of gradient descent farmed out on a network of eight 1.5 Mflop microcomputers (Inmos T800 TRAMS). We have experienced similar reliable convergence properties with a variety of algorithms — standard gradient descent; Newton Raphson; Simplex descents — none of which seemed to avail (either singly or in more clever combination) using the original objective function.

B Mathematical Details of Window Management

B.1 Upper Bound for $\hat{\vartheta}_{k-1}$

In Section 4.3.1 a worst case estimate for $\hat{\vartheta}_{k-1}$ was developed, its derivation follows from the fact that

$$\tilde{d}_k = \sum_{j=1}^{\tau_f} A_{\tau_r}^{k-j} n_N(\tau_r j) - A_{\tau_f} \Theta^{-1} \left[\begin{array}{c} \hat{C}_{k-1} n_S(k-1) \\ \hat{C}_k \left(n_S(k) + n_N(k-1)\right) \end{array} \right].$$

Noticing that

$$\begin{array}{l}
\|\hat{p}_k - d_k\|_M = \|\tilde{d}_k - \tilde{p}_k\|_M \\
= \| \left(A_{\tau_r} + G\hat{C}_k\right) \tilde{p}_{k-1} + n_{k-1} \\
+ \sum_{j=1}^{\tau_f} A_{\tau_r}^{k-j} n_N(\tau_r j) \\
-A_{\tau_f}\Theta^{-1} \left[\begin{array}{c} \hat{C}_{k-1} n_S(k-1) \\ \hat{C}_k \left(n_S(k) + n_N(k-1)\right) \end{array} \right] \|_M \\
\geq \frac{1}{\|(A_{\tau_r}+GC)^{-1}\|_M} \vartheta_{k-1} - \nu_\Delta(\rho_{k-1}, \rho_{k-2}),
\end{array}$$

where

$$\begin{array}{c}
\nu_\Delta(\rho_{k-1}, \rho_{k-2}) \stackrel{\triangle}{=} \|n_{k-1}\|_M + \alpha\tau_f\nu_N \\
+\frac{\alpha}{\|\Theta\|_M} \left(\nu_N + \|n_S(\rho_{k-1})\|_M + \|n_S(\rho_{k-2})\|_M\right),
\end{array}$$

we are led to define a worst case estimate for ϑ as

$$\hat{\vartheta}_{k-1} \stackrel{\triangle}{=} \left[\|\hat{p}_k - d_k\|_M + \nu_\Delta(\rho_{k-1}, \rho_{k-2})\right] / \bar{\alpha}. \tag{36}$$

B.2 Bounded Coupled Dynamics for ρ_k and ϑ_k

The bounded coupled dynamics for ρ_k and ϑ_k used in Section 4.3.3 is constructed by first approximating the appearance of ρ in n_k and ν_Δ to first order (13). This results in

$$\begin{array}{ll}
\|n_k\|_M & \leq \gamma(\nu_0 + \nu_1\rho_{k-1}) + \nu_N \\
\nu_\Delta(\rho_k, \rho_{k-1}) & \leq (1 + \alpha\tau_f)\nu_N + \gamma(\nu_0 + \nu_1\rho_k) \\
& +\frac{\alpha}{\|\Theta\|_M} \left(\nu_N + 2\nu_0 + \nu_1\rho_k + \nu_1\rho_{k-1}\right) \\
& = (1 + \alpha(\tau_f + 1/\|\Theta\|_M)\nu_N + (\gamma + 2\frac{\alpha}{\|\Theta\|_M})\nu_0 \\
& \nu_1 \left(\gamma + \frac{\alpha}{\|\Theta\|_M}\right) \rho_k + \nu_1 \frac{\alpha}{\|\Theta\|_M} \rho_{k-1}.
\end{array}$$

The coupled dynamical inequalities in question now may be written

$$\begin{array}{ll}
\vartheta_{k+1} & \leq \lambda_k\vartheta_k + \nu_1\rho_{k-1} + \gamma\nu_0 + \nu_N \\
\rho_{k+1} & \leq \alpha\left(\tau_f\nu_N + \gamma(\nu_0 + \nu_1\rho_{k-1}) + \nu_N\right. \\
& \left. +\frac{\lambda_k^2}{\bar{\alpha}} \left[\vartheta_k + 2\nu_\Delta(\rho_k, \rho_{k-1})\right].\right)
\end{array}$$

References

[1] Peter K. Allen, Aleksandra Timcenko, Billibon Yoshimi, and Paul Michelman. Trajectory filtering and prediction for automated tracking and grasping of a moving object. In *IEEE Int. Conf. Robt. Aut.*, pages 1850–1856, 1992.

[2] P. Anandan. Measuring visual motion from image sequences. Technical Report COINS-TR-87-21, COINS Department, University of Massachusetts, 1987.

[3] R. L. Andersson. *A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control.* MIT, Cambridge,MA, 1988.

[4] D. J. Bennett, J. M. Hollerbach, and D. Geiger. Autonomous robot calibration for hand-eye coordination. In *International Symposium of Robotics Research*, 1989.

[5] M. Bühler, D. E. Koditschek, and P.J. Kindlmann. A family of robot control strategies for intermittent dynamical environments. *IEEE Control Systems Magazine*, 10:16–22, Feb 1990.

[6] P. I. Corke. Video-rate robot visual servoing. In Koichi Hashimoto, editor, *Visual Servoing — Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback.* World Scientific, 1993.

[7] Ernst Dieter Dickmanns and Volker Graefe. Applications of dynamic monocular machine vision. *Machine Vision and Applications*, pages 241–261, 1988.

[8] Ernst Dieter Dickmanns and Volker Graefe. Dynamic monocular machine vision. *Machine Vision and Applications*, pages 223–240, 1988.

[9] Armando Fox and Seth Hutchenson. Exploiting visual constraints in teh synthesis of uncertainty-tolerant motion plans i: The directional backprojection. In *IEEE Int. Conf. Robt. Aut.*, pages 1:305–310, May 1993.

[10] Armando Fox and Seth Hutchenson. Exploiting visual constraints in teh synthesis of uncertainty-tolerant motion plans ii: The nondirectional backprojection. In *IEEE Int. Conf. Robt. Aut.*, pages 1:311–316, May 1993.

[11] Gregory D. Hager, Wen-Chung Chang, and A. S. Morse. Robot feedback control based on stereo vision: Towards calibration-free hand-eye coordination. In *IEEE Int. Conf. Robt. and Aut.*, page (to appear), San Diego California, May 1994.

[12] Bijoy k. Ghosh, Mrdjan Jankovic, and Y. T. Wu. Some problems in perspective system theory and its application to machine vision. In *Int. Conf. on Intelligent Robots and Systems*, pages 139–146, July 1992.

[13] D. E. Koditschek and M. Bühler. Analysis of a simplified hopping robot. *International Journal of Robotics Research*, 10(6), Dec 1991 .

[14] Brad Nelson and Pradeep K. Khosla. Integrating sensor placement and visual tracking strategies. In *Third Int. Symp. on Experimental Robotics*, 1993.

[15] Brad Nelson, N. P. Papanikolopolos, and P. K. Khosla. Visual servoing for robotic assembly. In Koichi Hashimoto, editor, *Visual Servoing — Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*. World Scientific, 1993.

[16] N. P. Papanikolopolos and P. K. Khosla. Adaptive robotic visual tracking: Theory and experiments. *IEEE Transactions on Automatic Control*, 38(3):429–445, 1993.

[17] A. A. Rizzi and D. E. Koditschek. Further progress in robot juggling: The spatial two-juggle. In *IEEE Int. Conf. Robt. Aut.*, pages 3:919–924, May 1993.

[18] A. A. Rizzi and D. E. Koditschek. Toward the control of attention in a dynamically dexterous robot. In Koichi Hashimoto, editor, *Visual Servoing — Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*. World Scientific, 1993.

[19] Alfred Rizzi and Daniel E. Koditschek. Preliminary experiments in robot juggling. In *Proc. Int. Symp. on Experimental Robotics*, Toulouse, France, June 1991. MIT Press.

[20] Alfred A. Rizzi and D. E. Koditschek. Progress in spatial robot juggling. In *IEEE Int. Conf. Robt. Aut.*, pages 775–780, Nice, France, May 1992.

[21] Alfred A. Rizzi and D. E. Koditschek. Progress in spatial robot juggling. In *IEEE Int. Conf. Robt. Aut.*, pages 775–780, Nice, France, May 1992.

[22] Alfred A. Rizzi, Louis L. Whitcomb, and D. E. Koditschek. Distributed real-time control of a spatial robot juggler. *IEEE Computer*, 25(5), May 1992.

[23] J. L. Synge and B. A. Griffith. *Principles of Mechanics*. McGraw Hill, London, 1959.