July 1993

Toward the control of attention in a dynamically dexterous robot

Alfred A. Rizzi
University of Michigan

Daniel E. Koditschek
University of Pennsylvania, kod@seas.upenn.edu

NOTE: At the time of publication, author Daniel Koditschek was affiliated with the University of Michigan. Currently, he is a faculty member in the Department of Electrical and Systems Engineering at the University of Pennsylvania.

Toward the control of attention in a dynamically dexterous robot

Abstract
In the recent successful effort to achieve the spatial two-juggle - batting two freely falling balls into independent stable periodic vertical orbits by repeated impacts with a three degree of freedom robot arm, the authors have found it necessary to introduce a dynamical window manager into their real-time stereo vision. This paper describes these necessary enhancements to the original vision system and then proposes a more formal account of how such a feedback based sensor might be understood to work. Further experimentation will be required to determine the extent to which the analytical model explains (and might thus be used as a tool to improve) the performance of the system presently working in the laboratory.

NOTE: At the time of publication, author Daniel Koditschek was affiliated with the University of Michigan. Currently, he is a faculty member in the Department of Electrical and Systems Engineering at the University of Pennsylvania.

Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems Yokohama, Japan July 26-30, 1993

Toward the Control of Attention in a Dynamically Dexterous Robot*

A. A. Rizzi †and D. E. Koditschek ‡
University of Michigan, Artificial Intelligence Laboratory
Department of Electrical Engineering and Computer Science

Abstract

In our recent successful effort to achieve the spatial two-juggle — batting two freely falling balls into independent stable periodic vertical orbits by repeated impacts with a three degree of freedom robot arm — we have found it necessary to introduce a dynamical window manager into our real-time stereo vision. This paper describes these necessary enhancements to the original vision system and then proposes a more formal account of how such a "feedback based" sensor might be understood to work. Further experimentation will be required to determine the extent to which our analytical model explains (and might thus be used as a tool to improve) the performance of the system presently working in our laboratory.

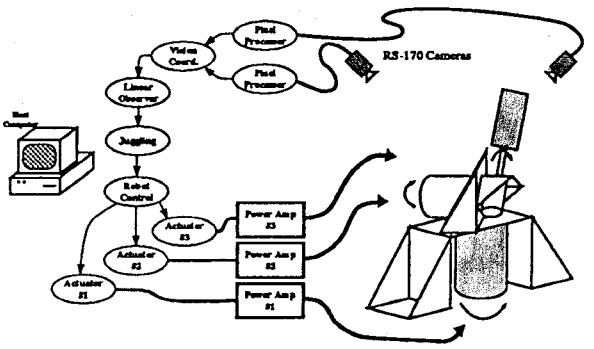

Figure 1: The Yale Spatial Juggler [10]

1 Introduction

This paper describes our recent efforts to enhance a real-time stereo vision system built as a "dynamical sensor" for our three degree of freedom Bühgler Arm [10]. The robot resulting from pairing our revised vision system with this arm has recently achieved a long-targeted milestone in our laboratory: the spatial two-juggle — the ability to bat two freely falling balls into independent stable periodic vertical orbits by repeated impacts with the arm [6]. There is no mechanically active component in this vision system. Nevertheless, we have found that there is already enough room for thought in getting its management right as to presage a wealth of important and novel problems that might be termed the "control of attention." Such problems are sure to arise in the planning and control of more general dynamically dexterous machines. The present paper is intended as an exploration of how to pose and think about such problems in a particularly simple case.

The term "dynamical sensor" is intended to convey our relatively narrow interest in programmable camera systems as a means of closing loops that result in the manipulation of the physical world. Such sensors must be dynamical in two different ways. First, the computational model of the environment to be sensed will stress the geometry of its dynamics rather than the geometry of its shape. Second, and perhaps more fundamentally, the computational strategy used in deployment will stress the connection between the sensor's dynamical role in a feedback loop and the quality of its measurements. Dynamical sensor management becomes a control problem. For example, since there are hard real-time constraints to meet, the management of resources must stress considerations of update and latency over mere throughput. Moreover, the process of extracting a little information from a lot of data is driven toward the minimum that will suffice for the task at hand rather than striving for the most that might logically be had. Finally, when previously sensed data mediates the collection of new information, a stability problem may result.

The architecture of our original setup is briefly reviewed in Section 2. The recent success of the two-juggle could not have been achieved without the enhancements to the vision system that we describe in Section 3. Although the working enhancements were developed in an ad hoc manner and implemented through a process of empirical trial and error, we suspect that the resulting system (or, at least, a suitably polished version thereof) should admit a relatively simple formal description. In Section 4 we present our progress in rendering such a formal description with the hope of promoting a more principled approach to solving such sensory control problems when we encounter them in the future.

2 Juggling Apparatus

Our juggling system, pictured in Figure 1, consists of three major components: an environment (the ball); the robot; and an environmental sensor (the vision system). After briefly sketching the properties of the first two of these we describe the originally conceived vision system in this section. All of this material has been presented in greater depth in [9, 8].

Bühler *et al.* [4] proposed a novel strategy for implicitly commanding a robot to "juggle" by forcing it to track a reference trajectory generated by a distorted reflection of the ball's continuous trajectory. This policy, the recourse a "mirror law," amounts to the choice of a map m from the phase space of the body to the joint space of the robot. A

†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.

0-7803-0823-9/93/$3.00 (C) 1993 IEEE

robot reference trajectory,

$$r(t) = m(b(t), \dot{b}(t)), \tag{1}$$

is generated by the geometry of the graph of m and the dynamics of ball, $b(t)$. This reference trajectory (along with the induced velocity and acceleration signals) can then be directly passed to a robot joint controller. [1] In following the prescribed joint space trajectory, the robot's paddle pursues a trajectory in space that periodically intersects that of the ball. The impacts induced at these intersections result in the desired juggling behavior.

Central to this juggling strategy is a sensory system capable of "keeping it's eyes on the ball." We require that the vision system produce a 1 KHz signal containing estimates of the ball's spatial position and velocity (six measurements). Denote this "robot reference rate" by the symbol $\tau_r = 10^{-3} sec$. Two RS-170 CCD television cameras constitute the "eyes" of the juggling system and deliver a frame consisting of a pair of interlaced frames at 60 Hz, so that a new field of data is available every $\tau_f = 16.\bar{6} \cdot 10^{-3} sec$. The CYCLOPS vision system, described in [8, 5], provides the hardware platform upon which the data in these fields are used to form the input signal to the mirror law, (1). The remainder of this section describes how this is done.

2.1 Triangulation and Flight Models

We work with the simple projective stereo camera model,

$$\mathrm{p} : \mathbb{R}^3 \to \mathbb{R}^4$$

that maps positions in affine 3-space to a pair of image plane projections in the standard manner. 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,"

$$\mathrm{p}^\dagger : \mathbb{R}^4 \to \mathbb{R}^3,$$

such that $\mathrm{p}^\dagger \circ \mathrm{p} = id_{\mathbb{R}^3}$. We have discussed our calibration procedure and choice of pseudo-inverse at length in previous publications [9, 8].

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,

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

where s denotes the sampling period, $\tilde{a}$ is the gravitational acceleration vector, and $w_j \in \mathbb{R}^6$.

2.2 Sensory Management

Following Andersson's experience in real-time visual servoing [1] we 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, necessitates a visually structured environment, and we presently illuminate white ping-pong balls with halogen lamps while putting black matte cloth cowling on the robot, floor, and curtaining off any background scene. Thus, the "world" as seen by the cameras contains only one or more white balls against a black background. In the case that only one white ball is presented, the result of this simple "early vision" strategy is a pair of pixel addresses, $c \in \mathbb{R}^4$, containing the centroid of the single illuminated region seen by each camera.

Figure 2 depicts the sensor management scheme we had employed to obtain ball positions in support of the previously reported spatial one juggle [9]. Each camera is serviced by a pair of processors. A field from a camera is acquired in time τ_f by one of the pair while the other is busy computing its centroid. The necessary computations will take longer than the allotted time, τ_f, if more than about 1200 pixels are examined. Thus, the moments are taken over a small subwindow of 30 by 40 pixels centered at the pixel location corresponding to the centroid address of the previously examined field. The pair of image plane centroids, $c \in \mathbb{R}^4$, is delivered to the vision coordinator at field rate, and is between one and two fields old, depending upon how much time it takes to form the centroid.

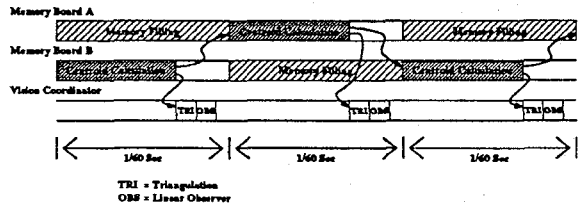

Figure 2: Timing Diagram for the Deployment of a Two Node Cyclops System in Support of Single Ball Sensing [10]

In summary, centroid data from one processor is passed over to the second whose window coordinates are adjusted accordingly. Note that this represents the active component in the sensing strategy upon which more attention will be focused below. The data is passed forward as well to the triangulation/observer processor. The two nodes then reverse roles, and the process repeats.

2.3 Signal Processing

Given a report of the ball's position from the triangulator, we employ a linear observer to recover its full state — positions and velocities. As described above, the window operates on pixel data that is at least one field old,

$$p_k = F^{-\tau_f}(w_k),$$

to produce a centroid. We use p_k as an "extra" state variable to denote this delayed image of the ball's state. 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. The data from the triangulator may now be written as

$$\bar{b}_k = \mathrm{p}^\dagger \circ W_k \circ \mathrm{p}(Cp_k). \tag{3}$$

Thus, the observer operates on the delayed data,

$$\hat{p}_{k+1} = F^{\tau_f}(\hat{p}_k) - G(C\hat{p}_k - \bar{b}_k), \tag{4}$$

[1] In the case of a one degree of freedom arm we found that a simple PD controller worked quite effectively [3]. In the present setting, we have found it necessary to introduce a nonlinear inverse dynamics based controller [11]. The high performance properties of this controller notwithstanding, our present success in achieving a spatial two-juggle has required some additional "smoothing" of the output of the mirror law described in a companion article [6].

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_{n_j} were available then it would be guaranteed that $\hat{p}_k \rightarrow p_k$. [2]

We provide the mirror law an appropriately extrapolated and interpolated version of these estimates as follows. The known latency is corrected by the prediction,

$$\hat{w}_k = F^{\tau_f + \iota_k}(\hat{p}_k),$$

where ι_k denotes the time required by the centroid computation at the k^{th} field. Subsequently, the mirror law is passed the next entry in the sequence,

$$F^{i\tau_r}(\hat{w}_k), (i = 1, ..., \tau_f - \iota_{k+1})$$

until the next estimate, $\hat{p}_{k+1}$ is ready.

3 Sensing Issues Arising from Actuator Constraints

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 the two-juggle experiments, we ignored this "detail" and happily ran with the open loop sensory management procedures used to obtain data (3). It soon became clear that these procedures could not be similarly transparent in the two-juggle. 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 [7] 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 much higher (longer flight times between impacts) and to bring the two balls much closer 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).

3.1 Modifications to the Sensing System

This section presents the intuitively generated modifications we have made to the originally designed sensing system described above. Individually, each of these "obvious hacks" represents a minor enhancement to the original system with little independent intellectual or engineering interest. However, getting them all to work in concert requires a greater amount of thought. Moreover, when combined, they significantly increase the capabilities of the robot: we have recently achieved the long targeted two-juggle behavior. Finally, their addition to the original sensor management system introduces the first hint in our work that controlling the machine's "state of attention" may be an important and fundamental problem in robotics.

3.1.1 Occlusion Detection

Bringing the two one-juggle tasks closer together in space greatly increases the potential for the balls to pass arbitrarily close together in a particular image resulting in an *occlusion* event. Handling such situations requires either the ability to detect and reject images containing occlusions, or to locate the balls reliably in spite of the occlusion. Our disinclination to pursue the second option relates to our interest in exploring robust and extensible algorithms suited to our computational resources. While a two-ball occlusion can be relatively easily disambiguated, more balls or more complicated shapes give rise to increasingly difficult and computationally intensive geometric problems. Instead, we prefer to make a very coarse (and presumably, more robust) decision concerning when an occlusion has occurred, and entrust to a dynamical model (the observer of Section 2.3) the precise localization of where either ball may be at any moment. As will be seen directly, this decision has consequences that set us out on the path of building a "dynamical sensor."

Since we have 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 second order moments as a simple and robust *occlusion detector.* Under well-structured lighting conditions, the "ballness" of an image is easily determined by putting thresholds around the the ratio of the eigenvalues of the matrix of the second order moments in conjunction with a test on the planar orientation its eigenvectors. When multiple balls appear in a single window — as determined by a data array that fails this second order moment test — the entire window of data is discarded and the observer simply integrates forward its present estimates. We presume that the results of such pure prediction will be more accurate than a computation based upon spurious centroid data.

An analogous line of reasoning supports our use of the zeroth order moment to characterize occlusions resulting from an out-of-frame or out-of-window event. A window of binary thresholded pixels with insufficient density is discarded as empty and the observer again updates its estimates on the basis of pure prediction. In the out-of-window event, the alternative strategy of re-examining the entire frame for the missing object is much too costly. In the out-of-frame event where a ball leaves the camera's field of view there is obviously no alternative to this strategy.

3.1.2 Observer Based Window Placement

In a situation where there are guaranteed to be regular occlusion events (because the balls are to be juggled high and close together), the policy outlined above of ignoring data from occluded windows severely compromises the effectiveness of the simple previously acceptable window placement manager. Recall from Section 2.2 that the original scheme simply used the centroid from the previous field as the window center in the next field. A spatial volume of roughly .1 meter diameter whose centroid is one field (.016 sec) old will not be likely to capture balls moving at speeds well in excess of 7 meters per second.

Instead, an obvious improvement results from using the estimates of the observer itself to place the windows.

[2] 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 (2) 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.

Namely, in the enhanced vision system, 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 locator has now become the output of a dynamical system internal to the robot whose inputs from the physical world we manage according to the decision process described above.

3.1.3 Impact Detection and Estimation

The two modifications described above have traded computational difficulty (simple geometric interpretation) for detailed dynamical knowledge (trusting the observer to correctly place the windows). However, the observer described in Section 2.3 is missing a model of a key dynamical feature in the life of the ball — the effect of the robot's impacts (u in (2)). If we drive the window manager with the output of the purely Newtonian observer then after the first impact the window center will continue to "fall" while the ball bounces up (with the relatively high velocity) and will almost certainly fail to lie within the next window — the ball is lost and the juggling stops.

In order to implement the observer with an enriched representation of the ball's dynamics we require both a model of impact and rather precise knowledge of the time the impact takes place. The former we have presented in [9]. The latter could be determined analytically in principle: starting with the assumption that the robot tracks its mirror law exactly (1); computing a position-velocity phase at contact; computing the induced effective impact. For reasons we have discussed at length in [2], our present mirror law constructions do not admit a closed form computation of the robot phase at contact. While numerical computation is a potentially feasible alternative, a predicted quantity will always be inferior to a sensed datum. Were the actual time of impact available, then a direct reading of the robot's joint space measurements could provide the sensory alternative. Thus, 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. The appropriate input, effectively a state change in the dynamical system (2) is calculated from the state of the ball and the robot at the time of the impact, and this is passed to the observer.

3.1.4 Window Size Adjustment

Although a central theme in this work concerns the advantages of trading a computationally intensive and brittle geometric model of the environment for a more robust dynamical model, there is no escaping the likelihood of error accumulation in either case. 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 but 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 sensor. First, noise inevitably corrupts the image frame 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, $\bar{a}$ in (2) 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 .25 seconds).

In the absence of a more principled approach to window area management, we have adopted the following strategy. Window area grows following any image plane measurement failure (i.e., an occlusion event). Window area shrinks following a valid measurement. The intuition is that we are capable of growing the windows large enough to compensate for the inevitable modeling error and reliably reacquire the ball either when it returns to the field of view or the occlusion ends. Conversely, after the observer has had a number of position inputs to process, we presume that the risk of losing the ball is outweighed by the potential advantage of gaining accuracy in the estimate from higher spatial resolution and minimizing the risk of further occlusions with the other ball/window.

3.1.5 Window Overlap/Prioritization

Of course, the larger the windows, the greater the likelihood of their overlapping and multiple balls being visible in individual windows. We have introduced an excision rule for removing intersecting regions from one window and assigning them exclusively to the other. Our rule weighs the cost of losing entirely a poorly tracked ball more heavily than that of corrupting the estimates of a relatively well tracked ball. This amounts to first looking for the things we know about in the image, blocking them out, and continuing to search for the remainder of the objects. Thus, we assign the windows a level of priority inversely corresponding to their size. The higher priority (smaller) window's pixels are excised from the moments computation of the lower priority (larger) window, but all of its pixels are used in the computation of its own moments.

In practice, this strategy seems to have the desired effect of not confusing a ball we are tracking well with one we have temporarily lost. That is, it avoids the spurious occlusion event caused by a well tracked ball (one we have seen in the recent past) entering a large window associated with a poorly tracked ball (one whose observer error has not yet grown small). More significantly, we have not yet introduced a means of discriminating between occlusions generated by out-of-frame versus window overlap conditions. For example it is not uncommon that a window overlap near the edge of the field of view is followed followed by one of the balls moving out of the field of view. Suppose the out-of-frame ball is assigned a higher priority than the ball still in view while the window overlap persists (that is, the in-view ball remains within the now enlarged window owned by the out-of-frame ball). The excision rule gives the pixels generated by the in-view ball to the out-of-frame ball's window, the window manager now starts to track the in-view ball, and the out-of-frame ball is lost. This sort of failure happens frequently enough that still more sophisticated window excision and overlap handling strategies than presently in place seem to be desirable.

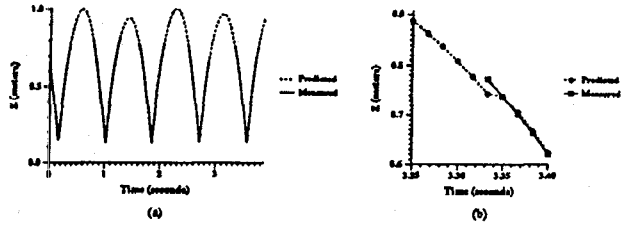

Figure 3: Measured and predicted (by the observer) ball heights for an out of frame juggling sequence (a), and an expanded view of a single recovery event (b).

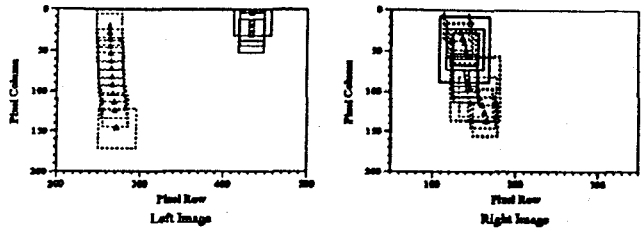

Figure 4: Left and Right image-plane tracks of a ball-ball occlusion event.

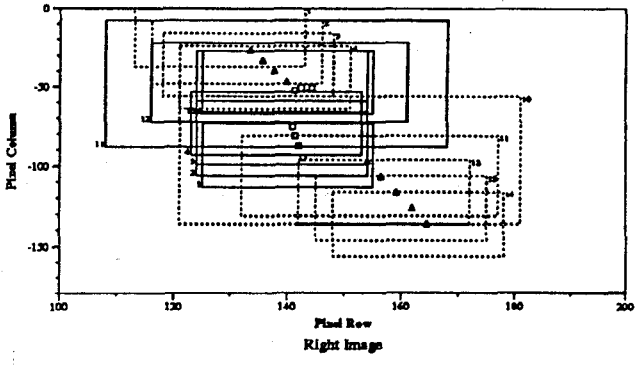

Figure 5: Expanded view of the left image-plane tracks showing the occlusion event.

3.2 Effect of the Modifications

We have recently achieved a functional two-juggle but have not yet logged more than a few dozen hits of both balls [6]. We are convinced that the sensing enhancements discussed above have significantly contributed to our recent success, and that their refinement will afford two-juggle performance comparable to our current one-juggle performance. Some documentation of this recent progress now follows.

3.2.1 Recovery from Out-of-Frame

As mentioned above, this set of modifications 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 3 (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 3(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 the ball was out of view. However since the measurement window has grown, this magnitude of error is readily accommodated.

3.2.2 Recovery from Ball-Ball Occlussions

Having recently succeed in presenting the vision system with two objects for a prolonged period of time, we have been been able to observe the occlusion events discussed above. Figure 4 and 5 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 5 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 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.

Although we have just begun to analyze data of this sort from our a working two-juggle we feel that a careful analysis of these events will allow for improved tuning of the window sizes and their rates of growth and shrinkage. Currently reliable recovery from these occlusion events remains the major obstacle to achieving sustained two-juggle performance we would consider comparable to that which we have been able to achieve with the one-juggle task.

4 Toward the Control of Attention

As more and more "enhancement modules" are added in the rather ad hoc fashion we have described, predicting and controlling their interactions becomes an increasingly difficult design problem. With the hope of developing a more principled approach to such design problems, we offer here a slightly more formal version of how to model and control the relevant sensor dynamics. It should be stressed that this formalism neither incorporates all nor cleaves faithfully in detail to any of the "enhancements" we presently employ. In contrast to those purely pragmatic measures adopted to "get on with the work," this re-examination is heavily weighted by considerations of analytical tractability. We are convinced that this interactive process of pragmatic building followed by theoretical reflection leading to further refined building, and so on, is the best way to advance the infant field of robotics.

Image plane windows that are too large will introduce unecessary noise through subsampling and time taken to compute the centroid. Larger windows will also have a higher probability of occluding when there are multiple targets to track. On the other hand, windows that are too small will be likely to loose their target with potentially catastrophic results. In this preliminary exploration, we focus on the

[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.

matter of how to place and size the windows in a rational manner.

4.1 The Window Management Variables as a "State of Attention"

The window manager controls the locus and extent of the image plane windows. Thus, we tentatively define a window's *state of attention* at some field interval, k, as the pair

$$a_k = \left(\hat{b}_k, \rho_k\right) \in \mathbb{R}^3 \times \mathbb{R}^+ \tag{5}$$

where $\hat{b}_k$ denotes an estimate of the spatial position of a falling ball, and where the positive scalar ρ_k is a measure of "certainty." With respect to a norm, $\|\cdot\|_M$, that will be defined below, a_k induces two windows on the two camera image planes including all stereo image pixel pairs, c having the property

$$\left\{c \in \mathrm{p}(\mathbb{R}^3) : \|\hat{b}_k - \mathrm{p}^\dagger(c)\|_M \leq \rho_k\right\}.$$

If enough of the pixels corresponding to the image of the ball pass through the imaging threshold to produce a sufficiently large zeroth order moment in the windows just defined, the first order moments will be passed to the triangulator to be interpreted as a spatial position. Otherwise, an "empty window" will be logged. 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

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

This notation immediately points up the *dynamics* intrinsic to the window management problem that appears at present 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.

For a suitable norm, we look back to the stabilized observer equations (4). 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

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

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

$$\|x\|_M \triangleq (x^{\mathrm{T}} M x)^{1/2}; \qquad \|A\|_M \triangleq \sup_{\|x\|_M = 1} \|Ax\|_M$$

For ease of exposition we introduce the notational conventions,

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

and assume, purely for further notational convenience, that the poles of the closed loop observer equation (4) have been placed on the real line with multiplicity two with the consequence that

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

4.2 Observer Errors from a Noisy Model

Clearly, 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 (2), 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 include 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. What seems more critical to emphasize is an incorporation in the noisy model of the particular effect of image plane geometry. For it is exactly the window size and consequent spatial resolution that is under control.

We substitute for (2) and (3) the system

$$\begin{aligned} w[(j+1)\tau_r] &= F^{\tau_r}(w(j\tau_r)) + n_N(j\tau_r) \\ p_{k+1} &= w[k\tau_f] \\ \bar{b}_k &= \hat{C}_k [p_k + n_S(\rho_{k-1})] \end{aligned} \tag{6}$$

It seems reasonable to take as a first crude model of the failings of the putative Newtonian free-flight model (2), n_N, a bounded deterministic sequence of uncontrolled inputs (perhaps generated via a map on the state space). The sensor noise introduced by thresholding a finite resolution image before computing moments is modeled by the function n_S. Because the resolution must decrease as the window magnitude increases in consequence of subsampling, n_S is nondecreasing in its argument. Because no subsampling is required for sufficiently small windows, n_S is a positive constant for small values of its argument. For present purposes it seems adequate to take n_S to be affine in ρ,

$$\|n_S(\rho_k)\|_M \leq \nu_0 + \nu_1 \rho_k \tag{7}$$

The deterministic output map, $\hat{C}_k$ returns the value $C = [I, 0]$ as in (2) when the body's image is in its assigned window, and vanishes otherwise:

$$\hat{C}_k \triangleq \begin{cases} C & : b_k \in \mathcal{N}(a_{k-1}) \\ 0 & : b_k \notin \mathcal{N}(a_{k-1}) \end{cases}. \tag{8}$$

We have determined in the face of an "empty window" to use simple extrapolation of the present estimate. Thus, the resulting observer takes the same form as (4) only with $\hat{C}_k$ (8) incorporated,

$$\begin{aligned} \hat{p}_{k+1} &= F^{\tau_f}(\hat{p}_k) + G(\bar{b}_k - \hat{C}_k \hat{p}_k) \\ \hat{w}(k\tau_f + j\tau_r) &= F^{\tau_f + \iota_k + j\tau_r}(\hat{p}_k), \\ &\qquad j = 0, 1, ..., \tau_f + \iota_{k+1} - \iota_k \\ \hat{b}_k &= CF^{\tau_f}(\hat{p}_k). \end{aligned} \tag{9}$$

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 window manager. The robot gets $\hat{w}(k\tau_f)$ as soon as it is formed: future predictions are made at the faster physical rate, τ_r. The window manager will make use of $\hat{p}_k$ in the form of $\hat{b}_k$ to handle the $(k+1)^{st}$ image.

There are now three distinct kinds of error, 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

$$\begin{aligned}\tilde{p}_{k+1} &= \left(A_{\tau_r} + G\hat{C}_k\right)\tilde{p}_k + n_k \\ n_k &\triangleq Gn_S(\rho_{k-1}) + n_N[(k-1)\tau_f].\end{aligned} \tag{10}$$

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

$$\begin{aligned}\vartheta_{k+1} &\leq \lambda_k\vartheta_k + \|n_k\|_M \\ \lambda_k &\triangleq \begin{cases} \bar{\alpha} < 1 & : b_k \in \mathcal{N}(a_{k-1}) \\ \alpha > 1 & : b_k \notin \mathcal{N}(a_{k-1}) \end{cases}\end{aligned} \tag{11}$$

and the condition on $\hat{C}_k$ and λ_k may now be expressed explicitly as

$$b_k \in \mathcal{N}(a_{k-1}) \Longleftrightarrow \|C^T(Cw[(k-1)\tau_f] - \hat{b}_{k-1})\|_M < \rho_{k-1}. \tag{12}$$

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

$$\begin{aligned}&\|C^T(Cw[(k-1)\tau_f] - \hat{b}_{k-1})\|_M \\ &\leq \|w[(k-1)\tau_f] - F^{\tau_f}(\hat{p}_{k-1})\|_M \\ &\leq \alpha\|\tilde{p}_{k-1}\|_M + \textstyle\sum_{j=1}^{\tau_f}(\alpha^{k-j}\|n_N[(k-2)\tau_f + j\tau_r]\|_M \\ &\leq \alpha(\vartheta_{k-1} + \tau_f\nu_N).\end{aligned} \tag{13}$$

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 (12) will hold.

The third sort of error 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, using arguments similar to those above,

$$\|\tilde{w}_k\|_M \leq \alpha^{\iota_k}(\vartheta_k + (\tau_f + \iota_k)\nu_N) \tag{14}$$

Since we prohibit the window manager from addressing more pixels than can be processed within the $16msec$ frame period, ι_k, the centroid computation time, increases by units of τ_r and saturates at the value τ_f:

$$\tau_r \leq \iota_k \leq \tau_f.$$

Thus, $\|\tilde{w}_k\|_M$ is a non-decreasing function of both ϑ and ρ.

4.3 Certainty Estimates from a Parallel Observer

Computations (13) imply 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. 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 get additional information concerning ϑ by comparing the two.

Using the invertibility of the observability matrix,

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

we may define a very different estimate of p of 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 for p 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, $n_S = n_N = 0$. In the present setting we have

$$\tilde{d}_k = \sum_{j=1}^{\tau_f} A_{\tau_r}^{k-j} n_N(\tau_r j) - A_{\tau_f}\Theta^{-1}\begin{bmatrix} \hat{C}_{k-1}n_S(k-1) \\ \hat{C}_k(n_S(k) + n_N(k-1)) \end{bmatrix}$$

and, noticing that

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

where

$$\begin{aligned}\nu_\Delta(\rho_{k-1}, \rho_{k-2}) &\triangleq \|n_{k-1}\|_M + \alpha\tau_f\nu_N \\ &+ \frac{\alpha}{\|\Theta\|_M}(\nu_N + \|n_S(\rho_{k-1})\|_M + \|n_S(\rho_{k-2})\|_M),\end{aligned}$$

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

$$\hat{\vartheta}_{k-1} \triangleq [\|\hat{p}_k - d_k\|_M + \nu_\Delta(\rho_{k-1}, \rho_{k-2})]/\bar{\alpha}, \tag{15}$$

guaranteeing that $\hat{\vartheta}_{k-1} \geq \vartheta_{k-1}$. For purposes of later analysis, note that

$$\hat{\vartheta}_{k-1} \leq \frac{\lambda_{k-1}}{\bar{\alpha}}[\vartheta_{k-1} + 2\nu_\Delta(\rho_k, \rho_{k-1})]. \tag{16}$$

4.4 Window Radius Dynamics for Bounded Estimator Errors

Equipped with a worst case estimate for ϑ, we are now in a position to adjust ρ. According to the previous calculations (13), a window radius management strategy that achieves the relation

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

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 (15) at stage k, we appeal to (11) and note that the desired relation is implied by

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

This demonstrates that the radius adjustment procedure

$$\rho_k = \alpha\left(\lambda_{k-1}\hat{\vartheta}_{k-1} + \|n_{k-1}\|_M + \tau_f\nu_N\right) \tag{17}$$

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

But there is now a question of observer convergence. For, 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 (11)). We must show that the coupled dynamical system (11), (17) remains bounded.

Approximating the appearance of ρ in n_k, ν_Δ to first order (7), we have

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

The coupled dynamical inequalities in question now may be written

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

4.5 Boundness of the State of Attention

Now in the coordinate system, $x \triangleq [\chi_1, \chi_2, \chi_3]^\intercal$, where $\chi_1(k) \geq \vartheta_k$ bounds the actual Lyapunov magnitude of (9) 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

$$\begin{aligned} x(k+1) &= Q_k x(k) + r \\ Q_k &\triangleq \begin{bmatrix} \lambda_k & 0 & \nu_1 \\ \frac{\alpha\lambda_k^2}{\bar{\alpha}} & \alpha\nu_1 g_1 & \alpha\nu_1 g_2 \\ 0 & 1 & 0 \end{bmatrix} \\ r &\triangleq \begin{bmatrix} r_1 \\ r_2 \\ 0 \end{bmatrix}, \end{aligned} \tag{18}$$

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 (17), 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.

5 Conclusion

The foregoing scheme is a reasonably faithful formalization of how the windows are placed in our present juggling system — the assignment of $\hat{b}_k$ (9). It is considerably less faithful to the way in which window radii, ρ_k, are presently determined. The chief advance in our thinking represented by the proposed formal radius management scheme (17) is the manner in which measurement uncertainty is incorporated. Specifically, the idea of running a second state estimator in parallel with the traditional Luenberger observer in order to compute a plausible bound on the estimate's error magnitude appears to be new. Whether it is also effective will depend upon the relative magnitude of sensor noise as compared to that of the drift term in the unstabilized Newtonian flight model (2). Further experimentation will be needed to ascertain whether or not this is so.

References

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

[2] M. Bühler , D. E. Koditschek, and P.J. Kindlmann. A Simple Juggling Robot: Theory and Experimentation. In V. Hayward and O. Khatib, editors, *Experimental Robotics I*, pages 35–73. Springer-Verlag, 1990.

[3] M. Bühler, D. E. Koditschek, and P. J. Kindlmann. Planning and control of a juggling robot. *International Journal of Robotics Research*, (to appear), 1992 .

[4] 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.

[5] M. Bühler, N. Vlamis, C. J. Taylor, and A. Ganz. The cyclops vision system. In *Proc. North American Transputer Users Group Meeting*, Salt Lake City, UT, APR 1989.

[6] A. A. Rizzi and D. E. Koditschek. Further progress in robot juggling: The spatial two-juggle. In *IEEE Int. Conf. Robt. Aut.*, page (to appear), May 1993.

[7] A. A. Rizzi and D. E. Koditschek. Toward the control of attention in a dynamically dexterous robot. In Koichi Hashimoto, editor, *Visual Servoing — Automatic Control of Mechanical Systems with Visual Sensors.* World Scientific, 1993 (to appear).

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

[9] 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.

[10] 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.

[11] Louis L. Whitcomb, Alfred Rizzi, and Daniel E. Koditschek. Comparative experiments with a new adaptive controller for robot arms. *IEEE Transactions on Robotics and Automation*, (to appear), 1993.