



4-1-1989

The Cyclops Vision System

Martin Buhler

Disney, Martin.Buehler@disney.com


N. Vlamis

Camillo J. Taylor

University of Pennsylvania, cjtaylor@cis.upenn.edu

A. Ganz

Follow this and additional works at: http://repository.upenn.edu/ease_papers

 Part of the [Electrical and Computer Engineering Commons](#), and the [Systems Engineering Commons](#)

Recommended Citation

Martin Buhler, N. Vlamis, Camillo J. Taylor, and A. Ganz, "The Cyclops Vision System", *Proc. North American Transputer Users Group Meeting* . April 1989.

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/ease_papers/787

For more information, please contact repository@pobox.upenn.edu.

The Cyclops Vision System

Abstract

Cyclops is a distributed real-time vision system. It is "real-time" as for most vision tasks, it can be configured with enough processing nodes as to allow an update rate of 60 Hz with a maximum latency of 1/30s. This allows the system to be used directly as a feedback sensor for motion control. Even though Cyclops was built originally for tracking objects in 3D at 60Hz, it offers great flexibility. It can be configured to attack many vision tasks at much higher rates than was previously possible with systems that are up to an order of magnitude more expensive.

For more information: [Kod*lab](#)

Disciplines

Electrical and Computer Engineering | Engineering | Systems Engineering

The Cyclops Vision System ¹

M. Bühler, N. Vlamis, C. J. Taylor, and A. Ganz

Center for Systems Science
Yale University, Department of Electrical Engineering
April 1989

Abstract

Cyclops is a distributed real-time vision system. It is "real-time" as for most vision tasks, it can be configured with enough processing nodes as to allow an update rate of 60 Hz with a maximum latency of 1/30 s. This allows the system to be used directly as a feedback sensor for motion control. Even though Cyclops was built originally for tracking objects in 3D at 60Hz, it offers great flexibility. It can be configured to attack many vision tasks at much higher rates than was previously possible with systems that are up to an order of magnitude more expensive.

Introduction

The development of Cyclops was motivated by specific needs for real time vision in robotics. A line of research in the Yale Robotics Lab addresses the understanding of controlled impacts with the goal of improving the dynamical dexterity of robots [7, 9, 8]. A simple planar juggling mechanism [10] which is controlled by a network of transputer based distributed real time control nodes [14, 11, 15, 12] is used for experimental study and implementation of theoretical results. Essentially the "robot" implements different juggling tasks by batting repetitively one or two pucks as they slide downwards on an inclined plane. As our tasks approach the limits of this apparatus, we are presently constructing a three degree of freedom juggling robot. The planar mechanism permitted simple and fast (1kHz) inductive position sensing of the pucks which carried internal oscillators. However, the spatial juggler will have to draw upon general stereo vision to sense the objects that are juggled.

Before we researched the market, we established the following performance criterion: The system should be able to track two (or more) objects in 2D at field rate (60Hz) and should lend itself to stereo operation for 3D. We assume structured lighting with white balls in front of a black background without any noise. The system should track the objects by successively locating them and computing their position via binary moment calculations on subwindows. The input is provided by a standard interlaced RS-170 video camera that is to be converted into an array of 512 by 512 eight bit pixels per full frame. Low level data filtering capability should be provided in hardware in order to free the CPU for higher level operations. Given our tight

¹This work has been supported in part by INMOS Corporation and the NSF Grants DMC-8552851 and DMC-8606213

budget constraints, the system (stereo) should cost less than \$30,000. This fits our conviction that in a control system the sensing elements — as well as the computing elements — should cost only a fraction of the system being controlled.

Furthermore, as the resulting data is used in closed loop robot control, the computational delay of the vision system (latency) is of critical importance for a successful implementation: In addition to the above mentioned update rate of 60 Hz, the latency should not exceed 33 ms. We feel that these are also representative minimum performance requirements for future real time robot vision systems.

To our surprise, an extensive market survey did not reveal a single system that clearly satisfied our requirements both in terms of performance and cost. In addition, many of the systems examined were very complicated and not easy to use. Most of them were apparently not targeted for real time vision. A system by Datacube [5] satisfied our performance requirements but exceeded the cost constraint. Interfacing the results to our control systems in a timely fashion posed another challenge. Motion Analysis builds a tracking camera [2] that satisfies our requirements as well. One camera tracks one or two objects (everything implemented in hardware) at 60 Hz. While the cost is almost within our constraints, this system has all the drawbacks of a “black box” approach: It offers no flexibility concerning for example object shape, tracking algorithm, noise rejection and increase in number of objects to track.

Often a homemade system can achieve better performance and flexibility for less cost when compared to commercially available alternatives. This is especially true in our situation, where the commercial options are very limited in a vision market that does not address real time issues broadly. By capitalizing on the latest VLSI technology, employing modular design and integrating our existing XP/DCS boards as video memory hosts, the project of designing and building a functional minimum configuration system (half of a stereo system) was completed by the second and third author as a semester project in a computer science class taught by the last author. Shortly thereafter it demonstrated tracking of two ping pong balls while using only half of the available processing time.

Cyclops started out as a system consisting of a digitizer board, a flow-through 2D filter board and two video memory boards (hosted by XP/DCS) which receive the alternate interlaced odd and even fields in a “1/2 frame ping-pong” fashion [13]. With minimum additional effort this scheme can be generalized in two ways. First, successive fields may be strobed into a sequence of up to eight different video memory boards. Second, one given field can be loaded into several video memory boards simultaneously. These two modes, termed “Scan Mode” and “Group Mode” respectively, allow the user to configure the system size and mode according to a given vision task and take advantage of the inherent parallelism in most vision applications. We thus created a very open architecture that initially was designed to satisfy a specific simple tracking task, but resulted in a real time vision system as flexible and general purpose as any of the commercial alternatives.

An additional valuable feature of this system is that it interfaces ideally to a distributed control network consisting of transputer based XP/DCS nodes, or to any other of the numerous

transputer based products available now. Furthermore it integrates directly in the standard INMOS software development environment (TDS) that facilitates the development, distribution and debugging of code for distributed transputer systems.

Hardware Description

One Cyclops system — two are needed for stereo vision — consists of four different units (three without the optional filter board) which are interconnected as shown in figure 1.

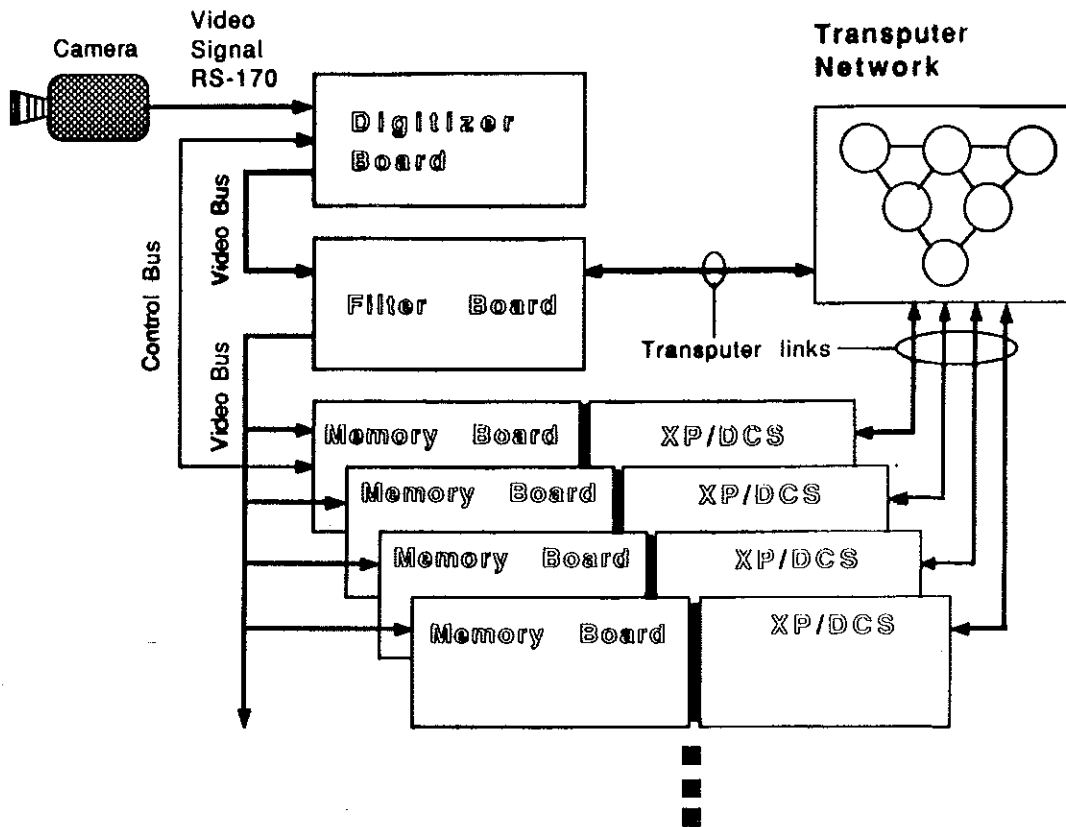


Figure 1: Cyclops functional blocks

Camera

Any camera that provides standard RS-170 video output can be used. In this standard, a frame is split up into two half-frames (even and odd fields) which are transmitted alternatively at a rate of 60 Hz [13]. In general the camera choice will be influenced by the specific application. For our object tracking task, we chose a CCD camera with a horizontal resolution of 600 lines. MOS cameras reset each pixel as it is being read, which causes the sampling interval to be different for

each pixel [6]. This complicates the reconstruction of the shape of a fast moving object. CCD cameras circumvent this problem by exposing all pixels simultaneously. In order to minimize the "smearing" effect that fast moving objects produce during the normal shutter speed of 1/60 s, this camera features an electronic shutter with a shutter speed of 1/2000 s. This reduces the distance an object with a speed of 6m/s can move during exposure time from 10 cm to 3 mm. Finally, for stereo operation the camera features a sync input that allow the synchronization of two or more cameras.

Digitizer Board

The digitizer board converts the analog RS-170 camera signal into the digital video bus. Each half-frame (field) is digitized as a 512 by 256 field of pixels, with eight bits (256 grey levels) per pixel. All this is accomplished by a novel hybrid integrated circuit [1] which simplifies this board considerably. The only additional circuitry consists of counters and switches in order to extract a desired active picture sub-region, and to generate a three bit address of the current target video memory board. In addition to the pixel data, the video bus carries the 9.83 MHz pixel clock, a data valid signal, the current memory board address, and the odd/even field indicator bit.

For increased flexibility, the digitizer board can be controlled via an optional Control Bus, which is connected to any one of the video memory boards. At any time, the digitizer board can be reset such that the next available field will be written to the (group of) video boards(s) with address zero or it can be requested to write the next field to any specific address.

Filter Board

This board adds powerful preprocessing capabilities to the Cyclops system. Its presence is transparent to both the digitizer and memory board. It introduces a time delay which is approximately equal to the total time of maximally six (filtered) rows. This delay is intrinsic to 2d filtering and not a hardware limitation. Several filter boards can be used in sequence to perform different kinds of filtering, for example to perform averaging on the first board before running an edge detection algorithm on the second.

The board consist primarily of five INMOS A110 "Image and Signal Processing Sub-systems" [3] and one INMOS T222 transputer [4] which dynamically loads coefficients into the A110 devices. The A110 is a one-dimensional (21 pixel length) or two-dimensional (3x7 pixel window) software configurable convolver/filter. Four of the five A110 devices used are operating in this mode to form a general user programmable 6x14 2D filter. The fifth device is taking advantage of its programmable line delays to delay the control signals in order to maintain correct timing with the pixel data after the filtering process.

The block diagram is given in figure 2. Besides miscellaneous buffer and address decoding

circuitry this diagram is very close to the actual schematics.

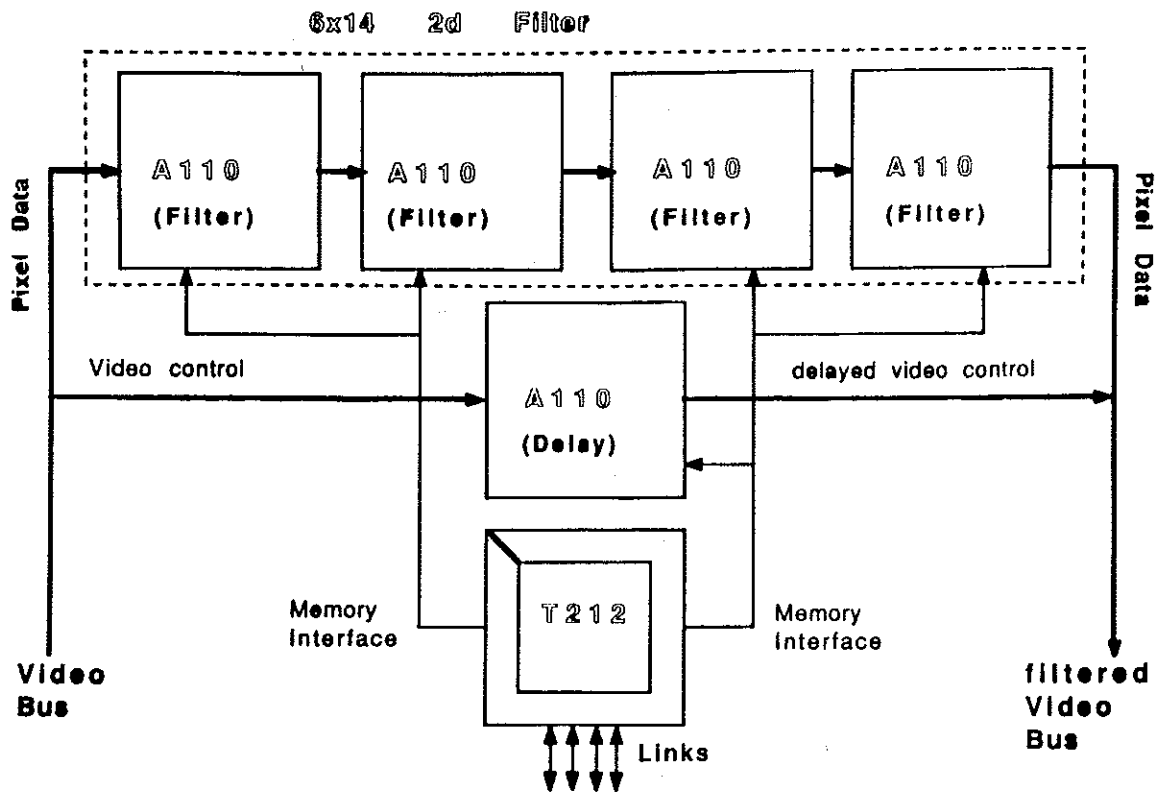


Figure 2: The filter board

Video Memory and Processor Boardset

The video memory board stores the digitized fields generated by the digitizer board. Once a field has been stored, the transputer on the processor board is interrupted to signal that video data is ready to be accessed. Each video memory board is hosted by an XP/DCS board. The latter is a simple transputer CPU board whose identity can be customized via a daughter board connector. Previously we have added a general purpose I/O daughter board and employed this boardset extensively for distributed real time motion control. A block diagram of this board is given below in figure 3. It consists basically of the video memory for one field (512 by 256), arbitration logic, and a comprehensive status and control interface to the mother board.

A **Terminal Count Register** contains the values of the address counters at the end of the last field that was stored. It shows how many rows and columns make up the active picture

area.

The **Status Register** contains the following signals: A *Contention Error Bit* is set whenever the transputer and the digitizer board try to access the video memory simultaneously. *Row Error* signals that two successive rows of the present field had different numbers of pixels. *Odd/Even* indicates if the video memory contains the even or odd lines of the video image. *Single Step Halt* is asserted whenever the board is in single step mode and contains a valid field. If the memory board receives an interrupt acknowledge from the transputer within a timeout period, *Frame Valid* is set, otherwise *Frame Missed* is set.

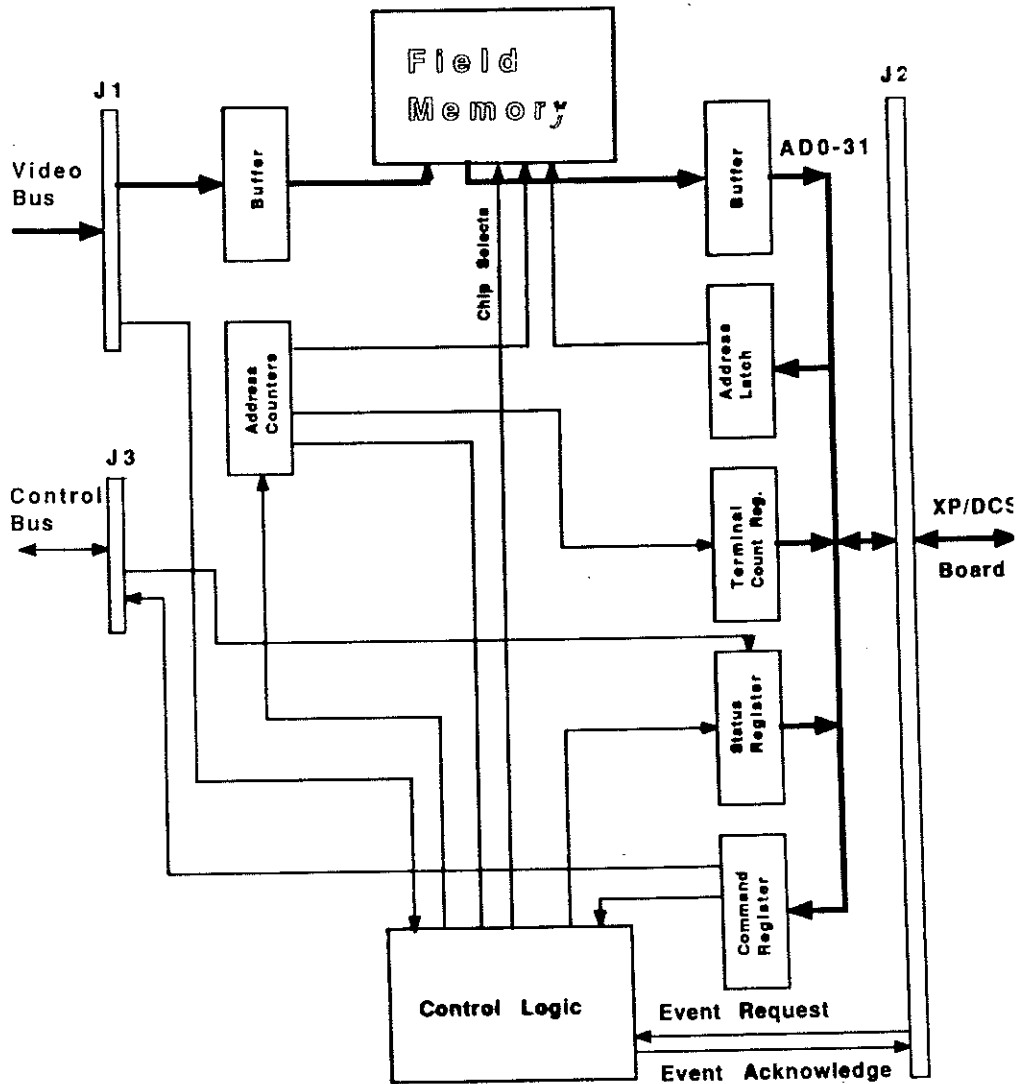


Figure 3: The video memory board

The **Command Register** comprises the signals that are passed on to the digitizer board and one bit that sets the memory board in single step mode.

In *Single Step Mode* the memory board is inhibited from loading any new field from the digitizer until instructed to do so, which allows the transputer arbitrary time to process a specific field. A new field can be requested by the transputer through software or manually by the user via a front panel push button.

Modes of Operation

Even though Cyclops satisfies a specific need, it is very flexible in its configuration and operation so as to allow its application to most vision tasks. In order to illustrate its generality, we will describe three different distribution schemes for the video fields and the implications for update rate and latency. By update rate we mean the rate at which results are computed by any of the processing boards on the video bus whereas latency refers to the total processing time from raw video data to end result.

First, consider the minimum (1/2 frame ping-pong) configuration consisting of two memory boards, which receive alternatively the odd and even fields. After a field has been loaded into board 1, processor 1 has 1/60 s to access, transfer and process it. While the next field is loaded, processor 1 has another 1/60 s (16.6 ms) to process, until the loading is completed. The timing for board 2 and processor 2 is analogous. Thus the total processing time and thus latency is 1/30 s (33.3 ms). However the update rate when the output of the two boards are combined is 60 Hz (16.6ms).

One can easily imagine how to generalize this idea and store successive fields in more than two different memory boards. In this mode, termed **Scan Mode**, successive fields are stored in up to eight different memory boards sequentially, as sketched in figure 4. In this fashion, the total processing time per processor can be increased up to 133 ms while maintaining an update rate of 60 Hz and without missing any field. This mode is very useful for image processing applications where results should be displayed graphically in an animated fashion for immediate visualization, for example in medical imaging. This mode is essential when the video images cannot be divided easily into spatially independent patches, on which different processors can work in parallel. If this is the case, the same video fields can be loaded into a number of memory boards simultaneously.

This latter mode we term **Group Mode** and show it graphically in figure 5. Depending on the application, each processor then picks a dynamic or static partition of the full field for processing. If only *one* group of video memory boards is used that receives the video fields simultaneously, latency is equal to update rate. Compared to the performance of *one* board, using *several* boards in this mode reduces latency. This is of critical importance in feedback control applications. In our object tracking task, *n* objects can be tracked by *n* video memory boards in group mode, where each board picks just the subwindow containing one object.

Mixed Mode combines scan and group mode. Successive fields are scanned into groups of boards as opposed to into single board as in scan mode. This reduces latency of the scan mode

while taking advantage of its high update rate.

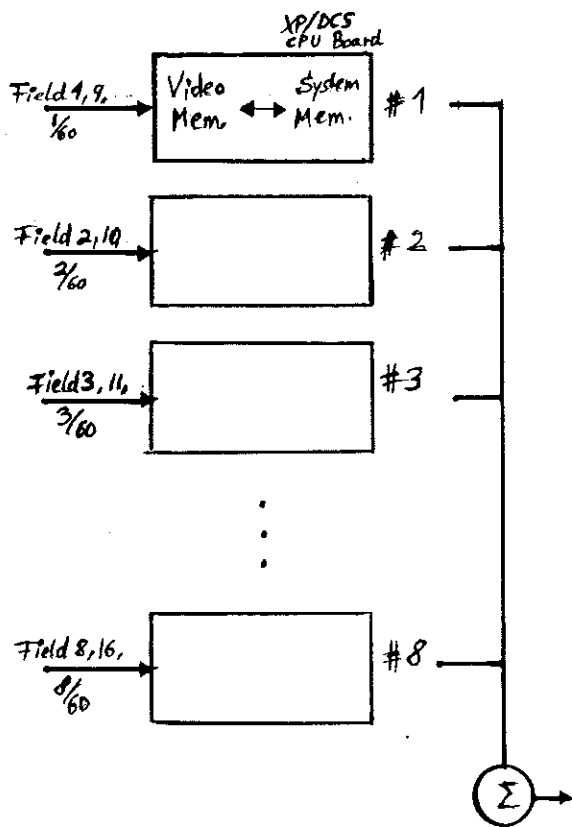


Figure 4: Scan Mode

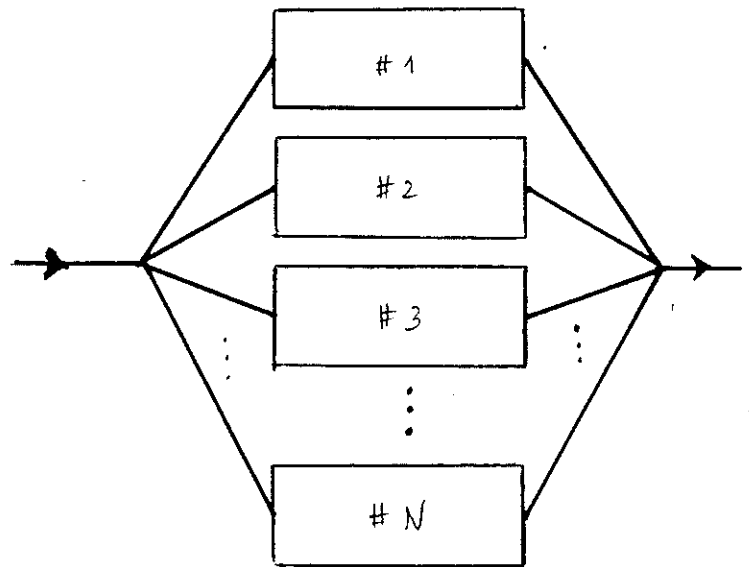


Figure 5: Group Mode

Conclusion

Cyclops has a variety of very attractive features for vision in general and for real time vision in particular. It is modular, can be built with moderate hardware effort, and would be competitively priced when produced commercially. The market value of homebuilt systems is somewhat difficult to assess. With parts cost for one cyclops system at about \$5000 we estimate that it

could be sold commercially between \$10,000 and \$15,000. Due to its parallel nature, its size can be adapted to the problem, and built up gradually, starting with a minimal configuration. Not only the size, but the operating mode can also be adapted. For example this can be effectively used to minimize the latency which is of critical importance for real-time-vision. As each parallel video memory node features a transputer, inter video memory communication as well as interfacing to an external parallel transputer network is simple and high bandwidth (20Mbps). Future industrial vision systems are likely to incorporate similar features.

The first version of Cyclops (without the filter board, which is currently under construction) was built and tested by the second and third author as a *semester project*. The critical ingredients which made this project a successful one were the existing design experience with the transputer in the Yale robotics lab, the availability of the XP/DCS CPU board, CCD cameras with high speed electronic shutter, single chip eight bit video digitizers (AD9502), INMOS 2D transform integrated circuits (A110), and affordable high speed static memories.

References

- [1] *AD9502 Hybrid RS-170 Video Digitizer*. Analog Devices, Norwood, MA, 1987.
- [2] *The Artist G110S Real Time Camera*. Motion Analysis Corporation, Santa Rosa, CA, 1988.
- [3] *IMS A110 Image and Signal Processing Sub-system*. INMOS Limited, Bristol, U.K., 1988.
- [4] *IMS T222 transputer*. INMOS Limited, Bristol, U.K., 1988.
- [5] *The MaxVideo System*. DATAcube Inc., Peabody, MA, 1986.
- [6] R. L. Andersson. *A Robot Ping-Pong Player*. MIT Press, Cambridge, MA, 1988.
- [7] M. Bühler and D. E. Koditschek. A prelude to juggling. In *26th IEEE Conference on Decision and Control*, pages (paper available from authors — not in proceedings), Los Angeles, CA, Dec 1987.
- [8] M. Bühler, D. E. Koditschek, and P. J. Kindlmann. A family of robot control strategies for intermittent dynamical environments. In *IEEE International Conference on Robotics and Automation*, page (to appear), AZ, May 1989.
- [9] M. Bühler, D. E. Koditschek, and P. J. Kindlmann. A one degree of freedom juggler in a two degree of freedom environment. In *Proc. IEEE Conference on Intelligent Systems and Robots*, page , Tokyo, Japan, Oct 1988.
- [10] M. Bühler, D. E. Koditschek, and P. J. Kindlmann. A simple juggling robot: theory and experimentation. In *The First International Symposium on Experimental Robots*, page (to appear), Montréal, Canada, Jun 1989.
- [11] M. Bühler, G. Unno, F. Levin, and D. E. Koditschek. Robots touch base. *Parallelogram*, (5):1-5, Aug 1988.

- [12] M. Bühler, L. Whitcomb, F. Levin, and D. E. Koditschek. A distributed message passing computational and i/o engine for real-time motion control. In *Proc. American Control Conference*, page (to appear), Pittsburgh, PA, Jun 1989.
- [13] Alan Hansford. *The AD9502 Video Signal Digitizer and Its Application*. Analog Devices, Norwood, MA, 1988.
- [14] F. Levin, M. Bühler, and D. E. Koditschek. The Yale Real-Time Distributed Control Node. In *Second Annual Workshop on Parallel Computing*, Oregon State University, Portland, OR, Apr 1988.
- [15] F. Levin, M. Bühler, L. L. Whitcomb, and D. E. Koditschek. A new computer board for distributed real-time motion control. *Electronic Systems Design*, 77-82, Feb 1989.