PennEyes
A Binocular Active Vision System

Brian C. Madden and Ulf M. Cahn von Seelen
{madden, cahn}@grip.cis.upenn.edu

GRASP Laboratory
Department of Computer and Information Science
University of Pennsylvania
3401 Walnut Street, Room 301C
Philadelphia, PA 19104, USA

December 1995

Abstract

PennEyes[1] is an experimental, binocular, three-dimensional tracking system. The goal was to design a high performance and extensible system using only off-the-shelf components thereby allowing limited resources to be concentrated on the development of vision and control algorithms rather than on the design of individual components. The capabilities of PennEyes will be reviewed as well as the rationale for its design.

1 Introduction

Much study has been done on the use of multiple cameras to construct three-dimensional representations of a real environment. Much study also has been done on controlling the position of cameras in those environments. While, at one time, even rudimentary accomplishments in these areas constituted major research challenges, much has been learned and that knowledge can be built upon. The goal attempted here was to design a positionable vision system from commercially available components. We wanted a tool to actively explore an arbitrary scene with a responsive binocular platform and, by doing so, obtain better representation of objects of interest. In particular, we were interested in obtaining quantitative performance measures of visual servoing. What we did not want to do was to fabricate the system from scratch out of glass, metal and chips. This report describes PennEyes, its components and the trades involved in its design.

There is always a trade in the design of an experimental system that balances the expenditure of available resources in the improvement of individual components against the expected improvement in performance of the integrated whole. At the outset, it is difficult to properly assess the increase in functionality any given component improvement will eventually afford the assembled system. Research on the components needed for tracking has often required a considerable amount of design and custom mechanical, optical and electronic fabrication (e.g., [Krot87, Pahl93, Shar93b, Will94])

[1] The current status of the system, together with technical reports and MPEG movies may be accessed through: http://www.cis.upenn.edu/~grasp/head/PennEyes/PennEyes.html.

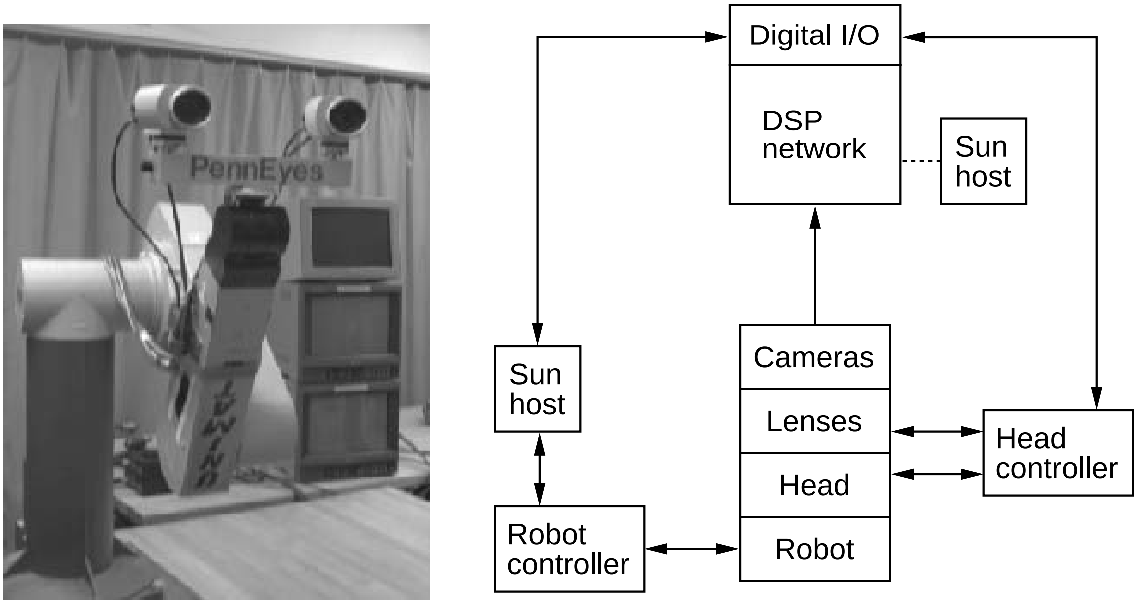

Figure 1: The PennEyes system and architecture. PennEyes is a head-in-hand system with a binocular camera platform mounted on a 6 dof robotic arm. Although physically limited to the reach of the arm, the functionality of the head is extended through the use of the motorized optics (10x zoom). The architecture is configured to rely minimally on external systems and communication. While program development can be done in a Unix environment on a workstation, the compiled code is loaded into the DSP network which runs stand-alone. The DSP network is directly connected to the head controller and is connected to the robot controller through a VME interface.

or reverse engineering (e.g., [Ferr91]). Even with these design successes, there is a small window of utility before advances in their constituent components renders them dated.

With PennEyes we have taken the approach that, at this time, the differential advantage of custom designs over commercially available components does not justify the additional resources and time necessary to fabricate components. After a review of the available products, we decided further that the advantage of buying a turn-key system does not justify the loss of flexibility and extensibility that modularity would provide. In essence, this philosophy is derived from the ability of a computer science laboratory such as ours to provide the greatest value-added in system integration. For us, the science is in the software. In accordance with these beliefs, we have attempted to assemble PennEyes with the best available off-the-shelf components. The final design (Figure 1) combined a two-axis BiSight camera platform (Transitions Research Corporation (TRC)), a Puma 560 robotic arm (Unimation/Westinghouse/Stäubli), a network of digital signal processors (TMS320C40/Texas Instruments) embedded in TIM-40 modules (Transtech Parallel Systems) and a VME-SPARC processor (Force Computers).

Apart from the decision to build a modular system with off-the-shelf components, there are several other factors that had considerable influence on the organization of the final system. These were decisions that made the configuration more in line with the physical and theoretical strengths of the GRASP Lab. *Mechanical:* The precision positioning afforded by a robotic arm was selected over the more indeterminate world of mobile platforms. The PennEyes system was designed to take advantage of the juxtaposition of two Puma 560 robotic arms (one to track in 3D and the other to provide independent 3D target motion). The latter provided a means of assessment of the performance of both the vision and the control algorithms, an objective assessment that unfortunately is not often present in computer science research. In addition, to be mountable on a

Puma 560 robotic arm, the binocular camera platform (pan drives, cameras and lenses) needed to weigh in the range of 2.5 kilograms (5.5 pounds). *Optics:* The ease of calibration and lower mass of fixed focal length lenses did not compensate for the inability to alter the field of view in a manner appropriate for the active vision paradigm. The use of motorized lenses (zoom, focus and aperture) offered an increase in functionality to an active vision tracking system over that afforded by fixed lenses that compensates for the increased weight, control and calibration complexity. *Electronics:* The most critical element in the design of the system was the image processing hardware. While improvements in the mechanical components of the system would make quantitative changes in the range, speed or precision of the tracking, the greatest opportunity for qualitative changes in tracking will come from increases in the computational capacity of the system. For some time to come, as the number of instructions that can be executed in 1/60th of a second increases, so will the complexity and abstractness of the targets that can be tracked as well as the variety of conditions under which they can be followed. A multiple instruction, multiple data (MIMD) DSP organization was decided upon as the best trade between performance, extensibility and ease of integration.

In the following sections that cover the mechanical, optical and electronic elements, their integration and the resulting performance of the PennEyes system, we will present evidence that supports these assertions.

2 Positioning

Among the earliest decisions was the type of positioning system for the cameras. One alternative was to place the cameras on a 4-axis head (two independent pan axes and head pan and tilt) mounted on a mobile platform (X and Z translation). The use of a mobile platform would allow the exploration of larger environments; however, factors such as slippage of the wheels would also increase the localization errors. While even an older robot such as the Puma 560 has a positioning repeatability of 0.1 mm and a working volume of a sphere nearly 2 meters in diameter, a mobile robot is largely constrained to planar translation and, in practice, can be localized to within several centimeters at best. When normalized by their respective precision, the span of the robotic arm is appreciably better. In addition, with commercially available components, it would be difficult to provide a vertical translation of the camera platform of a meter or more and still maintain stability of the moving platform. Only tilt could be easily accommodated.

The decision was made to accommodate a metrological approach, one which would stress quantitative measures of algorithmic performance and not the equally demanding task of maintaining robustness in a larger, uncertain domain. The availability of two Puma 560 robotic arms positioned 1.25 meters apart provided the potential for more precise quantification of tracking performance (Figure 2). This configuration allows one 6 degree-of-freedom (dof) arm to precisely position the head while the other independently provides a three-dimensional ground truth of known precision.

2.1 Robotic Arm

Although technically the 6 dof Puma 560 robotic arm should be able to arbitrarily position the head coordinate system any place within the workspace, problems can arise. In particular, singularities are problematical for real time tracking applications where the future path of the target is not known and path planning cannot be brought to bear. These singularities result from inopportune alignments of the joints that render the Jacobian noninvertible and therefore preclude the calculation of joint velocities from the desired Cartesian velocity of the end effector. At a cost of reducing the workspace volume, it is possible to configure the arm such that no possible path

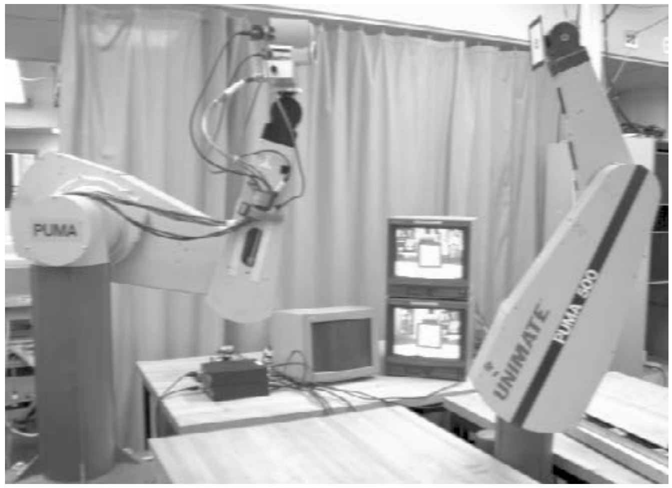

Figure 2: Puma Polka. Obtaining objective measures of tracking performance requires a precision target. With PennEyes, the proximity of a second 6 dof robot filled that need. A three-dimensional path with known precision can be repeatedly generated, allowing the comparison of different visual servoing algorithms.

can cause the arm to pass through a singularity (e.g., the configuration used in experiments on three-dimensional redundant tracking, see Figure 1). Further problems can arise when additional constraints are imposed such as maintaining a gravicentric orientation of the head. The sequence and type of joints be comes a factor. For example, head tilt must be mounted on head pan. Nonetheless, even when these impositions are required, a considerable workspace remains.

2.2 Binocular Head

Once the decision was made to go with a head-in-hand system, the only fixed design restriction was for the head to weigh within the payload envelope of the Puma 560 (approximately 5.5 pounds). In the beginning, fitting any optical functionality within this weight limit appeared beyond hope. It appeared as though a couple of pencil cameras with fixed lenses on a 6 cm baseline would be all that could be accommodated. Fortunately, a number of lightweight components became available at the right time and allowed a consortium of TRC and four universities to come up with a workable design and, eventually, a product. The resulting BiSight head is an example of a successful Small Business Innovative Research (SBIR) Award by TRC [Weim90, Weim92] in collaboration with the Universities of Pennsylvania, Rochester, Maryland and Massachusetts. The goal of the collaboration was to have a shared, commercially available hardware platform so as to promote software transfer among the various research programs.

The finished product, two independent pan axes with motorized lenses and CCD cameras, came in at 2.45 kg (5.4 pounds) (Figure 3). Given the rule of thumb that for a vergence system, good stereo resolution is provided over a distance approximately equal to ten times the camera baseline, the BiSight baseline of 25 cm is a good match to the working volume of the Puma arm. Even with the motorized lenses, the dynamic performance of the pan axes is exceptional (1000 deg/s peak velocity and 12,000 deg/s^2 peak acceleration). It is not that the peak velocity is often required (or even tolerated), it is that performance scales. The responsiveness of the head is reflected in the excellent tracking performance at moderate velocities. Lastly, the BiSight head met our requirement of a commercially available binocular camera platform.

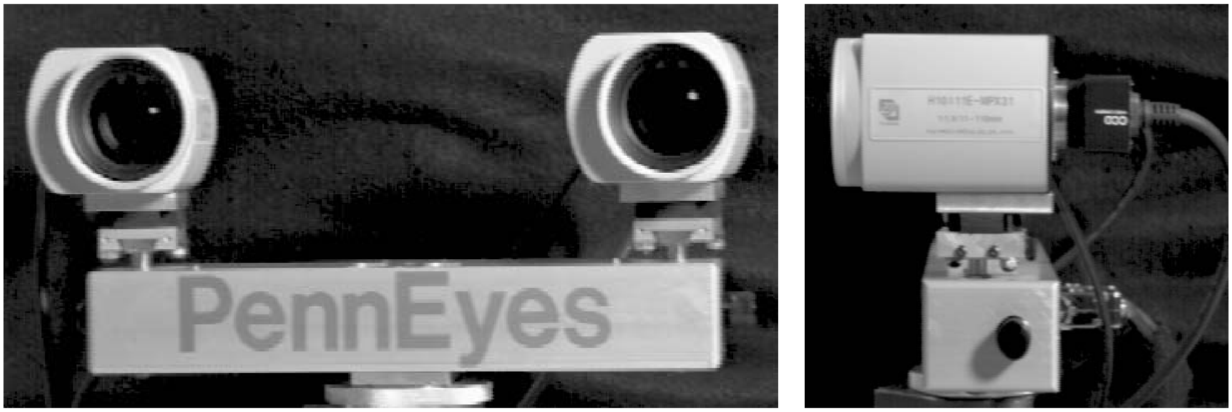

Figure 3: BiSight Head. The highest tracking performance is afforded by the independent pan axes on the binocular camera platform (1000 deg/s and 12,000 deg/s^2). Even with these lightweight imaging components, there was an appreciable cost for the added optical functionality. The combined weight of the lenses and cameras was 1.2 kg, approximately half the total weight of the head.

2.3 Head Optics

At the time the BiSight head was being designed, the range of commercially available motorized lenses was fairly small and those that did possess any functionality were quite heavy (typically 3.5 kg or more, e.g., Ernitec). These motors were made for the surveillance industry and they were made to survive in less than ideal environments. Precision and performance were secondary considerations. Just prior to the finalization of the head design, a lightweight motorized lens by Fujinon became available. The lens weighed 530 grams and had motorized focus (1.2 meters to infinity), zoom (11 to 110 mm) and aperture (F1.9). An additional benefit was the presence of feedback potentiometers on both zoom and focus.

Unfortunately, as with the earlier surveillance lenses, transit times for the zoom and focus were long (c. 6 s). In addition, with the low weight came plastic gearing. The degree to which these devices can maintain calibrated operation after repeated exposure to acceleration and vibration is not yet known. The original specification of the binocular camera platform called for manually positionable dovetail joints to allow the nodal point of each lens to be centered over the axis of rotation. This adjustment would avoid the translations associated with off-axis rotations and would partially simplify computations that must be done at field rates. As it turns out, unfortunately, the principal points of this type of zoom lens shift a considerable amount along the optical axis with changes in focal length. An adjustment capable of complete compensation for these shifts while maintaining rigidity would add considerable mass to the head; however, the small amount of manual positioning capability supplied with the BiSight head is useful in centering the mass of the camera/lens system over the pan axis.

2.4 Cameras

Another substantial savings in weight was made possible through the use of remote camera head sensors. The Sony XC-77RR black and white CCD cameras weigh just 65 grams. The amplifiers are 5 m down the video cable and do not need to be mounted on the arm. At NTSC frame rate (30 Hz), the 2/3 in sensor provides 756 pixels (11W by 13H micron photosites) per line, 485 lines (interlaced fields). The Sony cameras allow combinations of interlace/noninterlace and frame/field modes to be selected. Using noninterlaced field mode, a 242 line image can be obtained at 60 Hz.

In this configuration, the sensor integrates flux simultaneously at all pixel sites and then integrates the signal vertically over pairs of scanlines. Summing over pairs of scanlines restores the sensitivity lost by the reduction of the flux integration time associated with going from frame to field modes and reduces the amount of vertical aliasing by increasing the effective vertical dimension of the photosite on the sensor.

One tempting commercially available alternative was the use of color sensors. The use of color has been shown to greatly facilitate low level image processing such as real time segmentation. Single CCD color cameras were rejected because the nonuniformity of the color matrix would complicate other algorithms. Although there are remote head 3CCD color cameras similar to the Sony black and white XC-77RR, questions exist about the alignment of the three sensors. Both the initial congruence of the RGB sensors as well as the sustainability of that calibration in the presence of shock and vibration led us to believe that real time accurate color acquisition might be a problem. This consideration plus the additional complexity of handling three times the input bandwidth led us to decide on the black and white input for the present.

3 Control

A major design consideration was how to implement diverse control algorithms (e.g., PD, PID, Kalman filters, nonlinear control) in such a composite system. The goal here again was to use available resources for components whenever designing from scratch was unlikely to make significant improvements in performance.

3.1 Puma

For the Puma arms, the possibility existed to bypass the present controller and, by using the appreciable computational power of the DSP network, directly control the joint torques. This alternative, however, would require the creation of a dynamic model and determination of the associated parameter, a difficult task at best [Cork94b]. Instead, we controlled the robot with the public-domain RCCL/RCI (Robot Control C Library/Real-time Control Interface) package from McGill University [Lloy89, Lloy91].

RCCL/RCI allowed the Puma to be driven with C programs running under Unix on a SPARC-station IPX. The workstation communicates with the robot controller over a parallel interface VME card, via an SBus-to-VME converter. Another VME card, a counter module, generates high-priority hardware interrupts at regular intervals. The interrupts are serviced by a non-interruptible kernel-level process which computes the new setpoints for the arm and sends them to the host computer of the robot controller.

RCI provides the kernel additions that, with the help of the interrupt hardware, essentially transform Unix (i.e., SunOS 4.1) into a real time operating system. Under most circumstances, an RCI task will be executed regularly at the specified rate. A typical RCI rate is 50Hz; 100Hz seems to be the maximum that the workstation and the host computer in the robot controller are able to handle.

RCCL is a set of libraries that allows C programs to communicate with the RCI task through shared memory. The libraries offer different levels of control over the robot motion, namely interpolated trajectories in joint or Cartesian space, joint increments, or joint torques. On the side of the robot controller, setpoints or torque values are received by the host computer at the rate of the RCI task. They are transferred by the arm interface board to the six digital joint servo boards, which generate joint currents by executing a PID algorithm at approximately 1kHz (every 924μs). The controller can be set to compute increments for 8, 16, 32, 64 or 128 of these 1kHz intervals. These fixed intervals, together with the asynchronous operation of the cameras in field integration mode used for target position information, would cause difficulties for visual servoing if it were not for the re-entrant operation of the controller. With the controller set to update every 32 intervals and a 60 Hz rate of error computation, the receipt of the latest error signal causes a new setpoint target to be initiated at the next 1kHz clock. In this way the controller can seamlessly accommodate the new visual error signal. The operation of the robot controller is described in great detail in [Cork94a].

3.2 PMAC

The binocular camera platform has 4 optical (zoom and focus) and 2 mechanical (pan) degrees of freedom.[2] As part of the 2-axis BiSight system, TRC provided a PMAC (Programmable Multi-Axis Controller) 8-axis motion controller VME card (Delta Tau Data Systems). The PMAC is connected to the DSP network by a digital I/O interface. On the controller card, a Motorola DSP56001 digital signal processor runs the PMAC software, which is a mixture of a real time operating system and a command interpreter.

One of the strong advantages of PMAC is that it has an accessible architecture. Trajectory parameters, servo loop gains and even the DAC inputs are kept at documented locations in memory. All memory locations are, directly or indirectly, accessible and modifiable by the user. This openness permits greater control over the trajectory profiles.

PMAC was designed for generating high-precision preplanned trajectories for numerically controlled production machinery. For such applications, delays in the execution of motion commands and motion programs, due to trajectory planning and blending between successive moves, are not a problem. For real time reactive control, however, it is necessary to avoid these delays by driving PMAC at the servo level. The relative openness of PMAC's architecture makes such an approach possible though nontrivial to implement due to the difficulty of verifying a dynamic model of the head.

The PMAC provides direct control to the most responsive axes, the head pan (1000 deg/s peak velocity). While these axes can be used in combination with those on the robot to investigate the complexities of three-dimensional servoing, the two pan axes can be used alone as a platform to test the performance limits of simpler configurations (such as maintaining binocular fusion on rapidly moving targets along the horopter).

4 Image Processing

It is clear that tracking performance will continue to benefit from increased computational capability for some time to come. It is also true that there is a wide range of candidate systems to fill this need. The hardware solutions range from generic workstations to turn-key special-purpose vision architectures.

Workstations are desirable because they provide a comfortable development environment and there has been a history of continual improvement in workstation performance; however, there are two reasons why we decided against using workstations. First, the process scheduler in Unix-like operating systems decreases the priority of a process as its run time accumulates until the process is

[2] The aperture of each lens is also under computer control; however, there is no feedback available and the control is open-loop.

finally preempted. This behavior runs contrary to the demands of real time image processing, which requires the regular execution of CPU-intensive tasks. Although recent real time OS extensions (e.g., for Solaris 2.x and IRIX 5.x) provide alternatives to the conventional scheduler and promise a bounded response time to certain interrupts, their effectiveness in practice remains to be seen. The second reason for rejecting workstations is their restricted scalability. Upgrades and additions of processors can only effect a fixed increase in performance. For further performance gains, it will be necessary to integrate multiple workstations. The coordination of multiple workstations in a real time network raises further difficulties. While it is true that generic workstations will eventually have sufficient computing power to do visual servoing, it will first be accomplished by dedicated vision hardware.

At the other end of the spectrum are the turn-key vision systems such as image pyramids or stereo engines. These devices would be the ideal solution if one can be found that is flexible enough to accommodate a range of algorithms in active vision. We did not find this to be the case. Although these devices could often do one task very well, it was difficult to adapt them to other purposes. In addition, the systems were often proprietary black boxes. Internal details were not available, rendering them a risky platform upon which to base research.

In the range of special-purpose platforms for real time image processing, a common choice is the pipeline architecture. A popular example of this type is the MaxVideo system (Datacube). MaxVideo is a pipeline architecture that performs various linear and nonlinear operations in lock-step on an image sequence. However, pipeline architectures do not easily permit processes that have a nonuniform computational load or require extensive exception handling. Varying time demands do not match well the operational structure of a pipeline. Applications that involve higher-level visual processes made up of more than brute force convolutions require the flexibility of a MIMD architecture.

Our choice for a MIMD system was a network of digital signal processing modules based on the TMS320C40 DSP processor (Texas Instruments). The C40 processor is a well-documented commercial chip that offers high interconnectivity due to its six high-bandwidth communication ports (comports). Each comport has a dedicated DMA controller to free the CPU from I/O control. C40-based DSP modules are offered by a multitude of vendors. The modules are mounted on VME or PC motherboards that provide power, common reset, and basic comport connectivity between modules. The processing power of a C40 network can be increased by adding modules in the $1500 price range.

C40 code is usually developed on a Unix or PC host and downloaded on the C40 network for execution. We settled on a system based on VME motherboards and hosted by a VME SPARC board that runs the Parallel C development environment (3L Ltd.). The environment provides an optimizing C compiler and comprehensive libraries to generate the executables. A configurer packages the executables together with system tasks and a multitasking, multithreading microkernel into task images. Finally, the task images are downloaded by a distributed loader on the C40 network which then executes the code without any further intervention from the host.

The decision to rely on the C40 platform, however, has brought its own challenges. Using cutting-edge technology is never as comfortable as programming on a workstation. The learning curve is considerable as boards are often new and unproven and are rarely well-documented. Special-purpose boards such as digitizers use chips that require entire manuals. For the foreseeable future, image processing demands will consume all the available hardware performance. The scarcity of computational resources has unfortunate implications for software development. Time-critical routines either have to be coded in assembler, or the compiler-generated assembly code has to be inspected. Many convenient software features (multitasking, multithreading, high-level communication primitives) are of limited practical use due to their scheduling or function call overhead.

The C40 hardware technology is not without its own particular shortcomings. While the actual C40 modules conform to a standard (TIM-40), the motherboards are often incompatible between vendors. Although across-vendor interfacing hardware can be custom-designed, its costs and potential performance penalties tend to bind the customer to the same vendor when adding motherboards.

Some capabilities of the C40 processor that would be very useful for high-bandwidth applications have not been implemented by the board manufacturers. Currently available C40 modules permit only one-to-one comport connections between processors. The consequence is that if data transfers to different destinations have their source at the same location in memory, they have to be serialized. While the theoretical bandwidth of the C40 is quite high (up to 20MB/s per comport), it is in practice limited by slow memory and (on some kinds of motherboards) comport buffering. The serialization of transfers can therefore create bottlenecks. Such a situation arises when the 4-byte pixels (containing the bands of a color image or the gray values from multiple cameras) of an image sitting in slow digitizer VRAM have to be split up and distributed bytewise onto several other modules for parallel processing. The transfer time could be improved by a bus structure connecting multiple C40s. Although the processor supports a one-to-many connectivity, this feature has gone unused by the board manufacturers.

Another potentially useful feature of the C40 is the sharing of memory between C40 processors. Data transfer rates of 100MB/s can be achieved on the memory bus, which is five times as fast as the comport rate. However, shared memory has been implemented neither by module nor by motherboard manufacturers.

All these hardware and software difficulties notwithstanding, the high performance, flexibility, and control that the C40 technology affords for implementing real time image processing offers an appropriate balance for a real time research platform. The full PennEyes network comprises nine C40 modules. Figure 4 shows a configuration appropriate for a tracking application. The six comports on each module provide for a wide variety of configurations as well as for future expansion. Some of the modules are simple compute modules with fast memory while others have additional functionality. The following describes the various modules and discusses their capabilities and limitations.

4.1 Processing modules

Digitizers: The two TDM436 framegrabber modules (Transtech Parallel Systems) can digitize RGB or composite color video input. Alternatively, each module can digitize monochrome video from up to three sources simultaneously. The C40 processor on these modules only needs to initialize the line lock controller and A/D converter chips and various onboard registers. The actual digitization proceeds without intervention from the processor. The C40 can synchronize its operation with the video stream via polling or interrupts on the vertical and horizontal sync signals. Currently available digitizers suffer from the fact that they are equipped with slow dynamic memory as VRAM. This makes memory-intensive processing on the digitizer modules infeasible. One solution to this problem is to ship the images via DMA to C40 modules with fast static memory for processing. The TDM436 digitizers support this solution by performing the mask-and-shift operations necessary for separating the image bands in hardware.

Convolvers: The two VIPTIM convolution modules (National Engineering Laboratory, Scotland) each supplement a C40 with two 21-tap multiply-and-accumulate stages, achieving convolu-

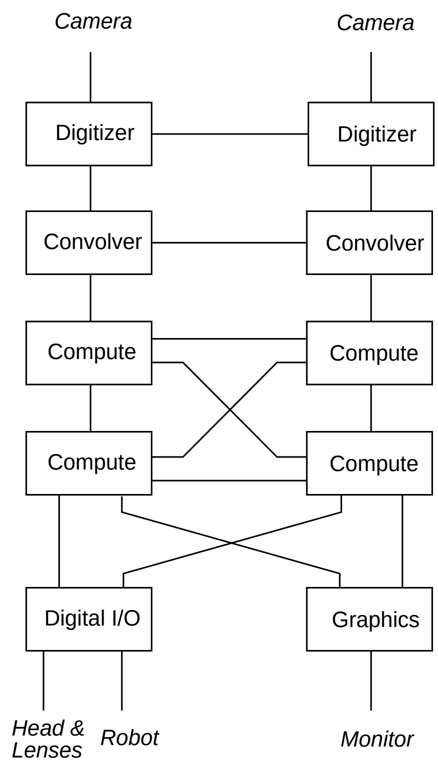

Figure 4: C40 Architecture. Beyond the basic computing power of the individual C40s, the performance of the network is enhanced by the ability to interconnect the modules with a fair degree of flexibility as well as the ability to locally store an appreciable amount of information. The former is made possible by using up to six comports on each module and the latter by several Mbytes of local storage. The block diagram shows an example configuration of special purpose and compute modules used for tracking. One advantage of this system is that additional modules (and capabilities) can be linked into the network without disturbing the pre-existing core.

tion with 42-tap one- or two-dimensional FIR filters at a rate of 10 Mpixels/s. This rate is an order of magnitude faster than the performance of the C40 alone. As with the digitizers, the memory is slow DRAM, which makes input and output a bottleneck on these modules.

Graphics: A helpful addition for debugging and visualization is the SMT304 graphics module (Sundance Multiprocessor Technology) which can produce analog color video output for VGA displays. A graphics engine placed between the memory and the RAMDAC relieves the C40 from basic drawing operations and block transfers. The RAMDAC supports overlay and cursor planes in addition to RGB color video.

Compute/Memory: Since the C40 can perform single-cycle integer and floating-point additions and multiplications, its real bottleneck is the memory access. Therefore, computation-intensive tasks are assigned to the four TDM407 modules (Transtech) that pair the C40 with fast SRAM (zero wait states for accesses within a page, single wait states on page misses).

Motherboards: Our decision on the motherboards and the host was influenced by two factors. The host of the C40 system was to fit into the workstation-dominated infrastructure of the lab. In addition, the system was to be capable of stand-alone operation at a remote site, e.g., aboard an unmanned ground vehicle. VME-based motherboards accommodated both of these requirements, allowing us to place the motherboards and the host, in the form of a VME SPARC board, into a single, portable chassis. The chassis then requires only power and interfacing to the cameras, the head, the robot and any monitors. A notebook computer suffices as the operator interface.

Host Board: The host of the C40 network is a VME SPARC board, SPARC CPU-5/CE (Force Computers). Under regular operation, the SPARC board is integrated into the lab's workstation and file server network over an ethernet connection, but a local disk permits stand-alone operation. The SPARC runs the 3L Parallel C development environment under Solaris 2.4.

Carrier Boards: Three TDMB428 motherboards (Transtech) with four C40 module sites each accommodate seven of our nine modules. They provide fixed comport connections between the sites as well as connectors to the unassigned comports for custom connections.

I/O Board: One motherboard, the TDMB424 (Transtech), is designed specifically for interfacing the C40 network to peripheral devices. This motherboard provides two C40 module sites and a VME interface through which the SPARC host downloads the code onto the network. In addition, the board contains two IndustryPack sites that are memory-mapped to the C40 module sites. IndustryPack is a growing standard for highly flexible and customizable I/O. A variety of I/O modules can be fitted to the IndustryPack sites. We used the sites for two digital parallel interfaces, one to the binocular camera platform and one to the robot arm.

Once the various optical, mechanical and electronic components are assembled, it remains only to ensure that the disparate elements work well together.

5 System Integration

In preliminary versions of the PennEyes system, the three principal subsystems were connected via Unix sockets on their Sun hosts. While this solution did not require any additional hardware, the obvious disadvantage was the indeterministic behavior of the ethernet connection and the Unix user-level processes necessary to transport the data through the sockets. Therefore, we decided to provide dedicated digital parallel lines between the C40 network and the head and robot. As mentioned above, one of the VME motherboards for the C40 network provides two IndustryPack sites. IndustryPack modules can be selected from a broad range of I/O functionality. These interfaces have sufficient bandwidth to easily accommodate the 60Hz rate of the visual error signals or even the 2kHz rate for direct control of the camera pan. We chose two parallel interface modules, each of which is configured for 32 bit-I/O lines and 8 handshake lines.

For the head controller, we added an I/O expansion VME card. This board connects directly to the PMAC VME card and provides 48 bit-I/O lines. This interface allows us to establish a direct connection between the C40 network and the PMAC card without going through the Sun hosts and the VME bus.

For the Puma, we had two alternatives to the socket connection. One solution was to install a second parallel interface VME card in the workstation and have an RCI real time task move the data between the new card and RCI's interface card. The other solution was to completely circumvent the host computer on the robot controller side and bring the parallel line directly into the arm interface board that normally transfers the data between the host computer and the robot's digital servo boards. This approach would have allowed us to provide setpoints or torque values to the robot at the Puma servo rate (1kHz). On the other hand, it would have required C40 device drivers to be written for the arm interface board.

In order to save development time, we decided on the first solution, even though it limits our setpoint updates to the rates achievable with RCI real time tasks. This solution still allows us to avoid Unix sockets and to go instead from the C40 network directly to the VMEbus where a regularly scheduled RCI task transfers the data to the robot. Together with the direct connection between the C40s and PMAC, we are able to provide deterministic communication links between the image processing system and the head and robot controllers.

5.1 Critical Issues

The performance of any modularly structured active vision system depends critically on a few recurring issues. They involve the coordination of processes running on different subsystems, the management of large data streams, processing and transmission delays, and the control of systems operating at different rates.

5.1.1 Synchronization

The three major components of our modular active vision system are independent entities that work at their own pace. The lack of a common time base makes synchronizing the components a difficult task. Even in the C40 network, the different modules use their own clock (although the nanosecond clock differential is insignificant given the millisecond time spans of the executed processes). In the following, we will discuss a variety of methods used to synchronize the operation of the C40s.

Synchronization among modules makes use of the Communicating Sequential Processes paradigm [Hoar85] implemented in the 3L Parallel C communication primitives. C40s communicate with each other over channels, which are mapped to comport connections. Sending or receiving a message blocks a C40 until the other processor has received or sent the message.

In some cases, an external signal can be used to synchronize independent hardware components. In our C40 network, the digitizers and the graphics module are slaved on the vertical sync of the genlocked cameras. The synchronization prevents beating between the update rate of the object position on the VGA display and the refresh rate of the VGA monitor.

To synchronize the transfer of the images from the digitizers, the transfer task is invoked by an interrupt derived from the vertical video sync signal. The minimum interrupt latency on a C40 is 8 cycles from the acknowledgment of the interrupt to the execution of the first instruction of the interrupt service routine. On a 50 MHz C40, this amounts to 320 ns, which is only a fraction of a video line.

We can use blocking synchronization methods in the C40 network without losing video fields because each C40 runs only a single, invariant task and the microkernel overhead is minimal and constant. Therefore each task always takes the same time, and the parallel processes interlock in a fixed order that keeps pace with the video input.

To interface time-critical tasks to processes that do not guarantee a response within a bounded time, non-blocking synchronization is necessary. Otherwise, the socket communication processes on the Unix hosts can hold up the image processing or the trajectory generation which have strict real time demands. Non-blocking synchronization is achieved by reading and writing the new data into a buffer shared between the time-bounded and the time-unbounded process.

Buffering introduces an unknown delay between the sending and the receipt of the data. In section 5.1.3 we discuss several ways to deal with latencies.

5.1.2 Bandwidth

Processing images requires working with high-bandwidth data streams. It is usually best to process images locally in fast memory, unless it is necessary to transfer images to other modules, e.g., to specialized hardware.

If data throughput becomes the bottleneck, solutions should be sought at the algorithmic level. For example, data rates can be kept low by working with a subsampled image or by limiting the processing to a smaller window.

Sometimes the total amount of computation per time can be decreased by *increasing* the sampling rate of the video stream. For example, if the frame rate is doubled, a tracked object can only move half as far between frames, and the search window can be halved. The amount of computation for a two-dimensional correlation search of the target, however, decreases quadratically. Therefore only a quarter of the computation has to be done, at twice the old rate, resulting in a saving of 50% (disregarding increased communications overhead).

High sampling rates also mean that temporal continuity constraints can be used to predict and decrease the search space. The work in [Dick90] draws much of its power from this approach to real time image processing.

5.1.3 Latency

Delays between the acquisition of a frame and the motor response to it are an inevitable problem of active vision systems. The flux integration time of the sensor can become a considerable factor in systems with short response time. The main latency, however, is usually caused by the image processing. Once the visual error is determined, an appropriate motor response is normally computed quickly.

Delays make the control more difficult because they can cause instabilities. It is a great advantage to make the inevitable delays invariant because then they can be incorporated into a plant model and used in a predictive control scheme. If this is not possible, an alternative is time-stamping [Shar93a]. Time-stamps on the visual error permit the control to adjust to the variable latency of the error signals by extrapolating the trajectory of the tracked object.

5.1.4 Multi-rate control

Active vision systems suggest by their very nature a hierarchical approach to control. The image processing component can generate a visual error at maximum rates of 25/30Hz (frame rate) or 50/60Hz (field rate) with conventional video cameras. The mechanical components of the system, on the other hand, typically have controllers that operate at rates of 500–2000Hz.

If the visual and mechanical control rates are one or more orders of magnitude apart, the mechanical control loops are essentially independent of the visual control loop. Provided that the actuators are responsive enough, they can be considered as black boxes that position the vision system as commanded by the visual error. Nonetheless, it is important to have explicit control over the actual shape of the commanded trajectories between setpoints. For example, if the positioning mechanism comes to a stop after each error signal, the tracking motion will become rough. For smoother tracking performance, velocity control should be implemented.

We have described a range of techniques that can adequately interface independent subsystems running at different rates. Part of our future research will attempt to quantify and optimize these control techniques.

6 Conclusions

Time is the problem. If the goal is to design real time tracking systems that scale – that cope with real world error and complexity, the time required to assemble the system will be considerable. We have found that development time can be reduced using commercially available components while achieving a high level of performance and functionality. PennEyes can attain velocities up to 1000 deg/s with zoom lenses and still be light enough to operate on the end of a robotic arm. We have also found, using modular design with off-the-shelf components, that it is possible to obtain good integration of communication and control without having the detailed level of control that comes with customized design.

We have designed PennEyes to be a responsive three-dimensional visual servo that is scalable. Scalable both in the sense that the results are applicable to real environments that contain vibrations and electrical noise, friction and latencies as well as in the sense that it can be extended by replacing and adding to its components.

6.1 Future Directions

Time is also a double-edged sword. While it renders work accomplished dated, it offers new technologies and capabilities. In the short period since the design of the PennEyes system, there have been many advances in the available components. Optical specialty houses are beginning to offer precision zoom lenses at a weight that makes their use practical. IndustryPack interfaces for the Puma controller are now sold that will allow higher rates of communication between the DSPs and the robot arm. Camcorder manufacturers are selling complete subassemblies (3CCD with 12x zoom) that include microprocessors, facilitating the eventual shift of more and more computation back to the sensor itself. Also in this vein, there are new intelligent sensors with both analog and digital computation available at the photosite and random access data transfer. It will soon be possible to extend the MIMD image processing network to incorporate the newer chips (e.g., the TMS320C80) and thereby take a step toward obtaining the hundreds of Gflops required for real time tracking of arbitrary targets under arbitrary conditions. With the modular design of PennEyes we expect to be able to use these advances in components to both replace and augment parts of the system with a minimum of disruption to existing capabilities.

All too often results are presented in the absence of contextual influences. In this report we have attempted to include both in order to give more meaning to the descriptions and to provide a reference for others confronted with similar decisions.

Acknowledgements

The equipment used in this work was supported by ARPA Grants N00014-92-J-1647 and DAAH04-93-G-0419; ArmyDAAL 03-89-C-0031PRI; NSF Grants CISECDA 88-22719, STC SBR-8920230, CDA-9121973, IRI 89-06770, ASC 91-08013, MSS-91-57156, and CISECDA 90-2253; NATO Grant 0224/85; A.I. duPont Institute, Barrett Technology Inc., duPont Corporation, General Motors and the Preservation Hall Trust.

References

[Cork94a] Corke, P. I. (1994) High-performance visual closed-loop robot control, PhD. Thesis, University of Melbourne, Australia.

[Cork94b] Corke, P. I. and Armstrong-Hélouvry, B. (1994) A search for consensus among model parameters reported for the PUMA 560 robot. In *Proc. IEEE Int. Conf. Robotics and Automation*, San Diego.

[Dick90] Dickmanns, E. D. (1990) Visual dynamic scene understanding exploiting high-level spatio-temporal models. *ICPR*, pp. 373–378.

[Ferr91] Ferrier, N. (1991) The Harvard binocular head. Technical Report 91-9, Harvard University.

[Hoar85] Hoare, C. A. R. (1985) *Communicating Sequential Processes.* Prentice Hall, Englewood Cliffs, NJ.

[Krot87] Krotkov, E. P. (1987) Exploratory visual sensing for determining spatial layout with an agile stereo camera system. PhD. Thesis, The University of Pennsylvania.

[Lloy89] Lloyd, J. and Hayward, V. (1989) *Multi-RCCL User's Guide and Reference Manual.* Centre for Intelligent Machines, McGill University, Montréal, Québec, Canada.

[Lloy91] Lloyd, J. (1991) *RCI User's Guide and Reference Manual.* Centre for Intelligent Machines, McGill University, Montréal, Québec, Canada.

[Pahl93] Pahlavan, K. (1993) Active robot vision and primary ocular processes. PhD. Thesis, The Royal Institute of Technology, Sweden.

[Shar93a] Sharkey, P. M. and Murray, D. W. (1993) Coping with delays for real-time gaze control. *SPIE Sensor Fusion VI.*

[Shar93b] Sharkey, P. M., Murray, D. W., Vandevelde, S., Reid, I. D. and McLauchlan, P. F. (1993) A modular head/eye platform for real-time active revision. *Mechatronics* **3(4)**, pp. 517–535.

[Weim90] Weiman, Carl (1990) Robot vision system based on log-polar image plane coordinates. SBIR Proposal, Phase I, Transitions Research Corporation.

[Weim92] Weiman, Carl (1992) Robot vision system based on log-polar image plane coordinates. SBIR Proposal, Phase II, Transitions Research Corporation.

[Will94] Willson, R. and Shafer, S. A. (1994) What is the center of the image? *JOSA A* **11**, pp. 2946–2955.