A miniaturized active vision system *

Benjamin B. Bederson †‡
Richard S. Wallace †
Eric L. Schwartz †‡§

†Vision Applications, Inc.
611 Broadway #714
New York, NY 10012

‡Courant Institute of Mathematical Sciences
New York University
251 Mercer Street
New York, NY 10012

§Computational Neuroscience Laboratory
Department of Psychiatry
New York University Medical Center
550 First Avenue
New York, NY 10016

November 13, 1991

Abstract

We have developed a prototype miniaturized active vision system whose sensor architecture is based on a logarithmic structured space-variant pixel geometry. This system integrates a CCD sensor, mini pan-tilt actuator, controller, general purpose processor and display. Due to the ability of space-variant sensors to cover large work-spaces yet provide high acuity with an extremely small number of pixels, space-variant active vision system architectures provide the potential for radical reductions in system size and cost. In this paper, we describe a prototype space-variant active vision system which performs such tasks as connected components analysis and motion tracking. The potential application domains for systems of this type include vision systems for mobile robots and robot manipulators, traffic monitoring systems, security and surveillance, and consumer video communications.

*Submitted to 11th ICPR specialty conference on Architectures for Vision and Pattern Recognition. Supported by DARPA/ONR #N00014-90-C-0049. Please address all correspondence to the authors at Vision Applications, Inc., 611 Broadway #714, New York, NY 10012.

Figure 1: Prototype miniaturized active vision system.

1 Introduction

We have developed a prototype miniaturized active vision system whose sensor architecture is based on a logarithmic structured space-variant pixel geometry [15]. This system integrates a CCD sensor, mini pan-tilt actuator, controller, general purpose processor and display [Figure 1]. Due to the ability of space-variant sensors to cover large work-spaces yet provide high acuity with an extremely small number of pixels, space-variant active vision system architectures provide the potential for radical reductions in system size and cost. In order to exploit these characteristics, a number of novel problems must be solved, including the design of small, low cost pan-tilt actuators, the production of custom VLSI sensors or embedded hardware emulations of space-variant sensors, and algorithmic methods for space-variant image processing, motor control, and visual attention. In this paper, we describe a prototype space-variant active vision system which may be connected to a host processor for applications development, or integrated as the visual module of a larger robotic system. The potential application domains for systems of this type include vision systems for mobile robots and robot manipulators, traffic monitoring systems, security and surveillance, and consumer video communications.

There have been a number of descriptions of the advantages to using space-variant, or logmap, architectures for video sensors [14] [16] [18] [19] [21] [26] [Figure 2]. We are currently working with Synaptics, Inc. to fabricate a CMOS logmap sensor, which is expected to be available within the next year. In the system which we describe here, we have emulated the logmap sensor by mapping the data from a conventional CCD into the logmap coordinates, using an embedded microcontroller and an optimized image warp algorithm. This emulation is currently running at 10 frames per second.

The CCD and lens assembly are mounted in a novel pan-tilt actuator. called a Spherical Pointing Motor (SPM) [5] which is capable of very fast, open-loop actuation of small camera heads [Figure 3].

As an architecture for image processing and computer vision, our prototype represents a significant departure from the types of large-scale parallel vision systems we and others have previously investigated (e.g. [8][12][24]). The new design is a small, loosely coupled network of embedded microcomputer modules, integrated directly with a sensor and robotic pan-tilt actuator. Moreover, our architecture has benefited from research into the space-variant nature of the human visual system [18][19].

Other researchers have developed a variety of active vision systems that mimic some biological functions such as pan, tilt, vergence and focus control, attentional algorithms and real-time tracking [1] [2] [3] [4] [6] [7] [9] [10] [11], but through implementing image processing at the low data rate of a logmap sensor, and exploiting our novel pan-tilt mechanism, our prototype demonstrates a significant subset of these capabilities with only modest processing power and inexpensive components. The two principles taken together, embedded system design and space-variant active sensing, enable us to flexibly develop new applications at a very low cost.

2 Prototype system

The prototype system consists of the emulated logmap sensor, (which will be replaced by a custom VLSI sensor in the near future), a mini pan-tilt mechanism, controller, general purpose processor and display. The controller consists of a camera driver, a 2 MIPS programmable microcontroller (Motorola MC68332), a video display driver and a 12 MIPS digital signal processor (Analog Devices AD2101). The actuator and camera is mounted to the chassis ($14 \times 22 \times 22$ cm) and connected by twisted-pair cables. The prototype is powered from a standard 110 Volt AC line, but uses less than 20 watts and could be battery powered.

2.1 Camera sensor

The camera sensor is a miniature commercially available CCD and lens assembly mounted in the pan-tilt actuator. The CCD image is converted to a logmap image containing less than 1500 pixels. Its maximum central resolution is 0.175 degrees per pixel and its horizontal field of view measures 33^o. The sensor has a fixed focus 4mm lens with a manually changeable aperture (3 sizes). Imaged objects more than 40mm away from the lens are in focus. The system outputs up to 10 frames per second and measures 256 gray levels per pixel. The camera head (CCD and lens assembly) measures only $8 \times 8 \times 10$mm.

The camera driver board is the interface between the camera sensor and the MC68332 microcontroller. This board provides timing signals to the sensor and converts analog sensor data to 8-bit digital data for the microcontroller.

Figure 2: A logmap (or log polar) image containing 1352 pixels.

Figure 3: The Spherical Pointing Motor and camera sensor. The camera head (CCD and lens assembly) measures only $8 \times 8 \times 10$mm. The SPM takes up a volume $4 \times 5 \times 6$cm.

2.2 Pan-tilt actuator

The pan-tilt actuator is a novel design using three orthogonal motor windings to achieve open-loop pan-tilt actuation of the camera sensor in a small, low-power package. The pan-tilt actuator is called the Spherical Pointing Motor (SPM). The SPM can orient the sensor through approximately 90^o pan and tilt, at speeds up to 500 degrees per second. Its size is $4 \times 5 \times 6$cm and its mass is 170 grams. The control currents in three orthogonal motor windings vary under pulse width modulation control. We have reported details of the SPM in a companion paper [5].

2.3 Microcontroller board

A Motorola MC68332 microcontroller controls the camera and SPM and runs some applications software. The MC68332 may be connected to a host processor for applications development, or to upload sensor data and to receive pan-tilt commands. The MC68332 is a 32 bit 16MHz microcontroller integrating peripheral controls, such as programmable digital control lines and timing signals, directly on chip. We connected 192K bytes RAM and 128K bytes EPROM to the microcontroller. The logmap data is output via a high speed serial interace, either to an external device such as a host computer or to the internal display driver.

We have implemented image processing demonstrations in the microcontroller ROM. A simple motion tracking program, which turns the camera to center the observed motion field, runs at 4 frames per second (4 fps). The ROM also contains a software automatic gain control (8 fps), a test pattern, a selectable frames per second control, image binarization (8 fps) and a motor scan sequence demo. Connected components analysis using a connectivity graph implementation [22], the slowest program, runs at 2 fps on the 2 MIPS M68332, illustrating both the generality of the programming model and the limitations of the microcontroller in the current prototype.

2.4 Digital signal processors

The Analog Devices 2101 digital signal processor (DSP) provides additional real-time image processing capabilities on logmap images. The DSP board might be programmed to control a robot hand, to perform real-time pattern recognition tasks, or even for voice output. The DSP obtains logmap image data from the MC68332 microcontroller in real-time via a high speed serial port. The DSP board combines an AD 2101 12 MHz DSP having two 2 MHz serial ports, 8K bytes internal RAM, 80K bytes external RAM, and 64K bytes boot EPROM.

An additional DSP board, identical to the first, functions as a video display driver, generating an RS-170 video output suitable for display on a standard TV monitor.

3 Programming and development model

3.1 Embedded system model

Each of the three microprocessors in our prototype is a general purpose computer, so the programming model bears great resemblance to that for embedded control systems on general purpose processors. One of the key elements in this model is a distinction between a development system and a target system. We initially developed our applications on a Sun workstation based system. This development system simulates the target system to a high degree: it combines a robotic pan-tilt actuator made by Klinger Scientific, a Sony XC-77RR camera and Analogic video digitizer, with software tools to simulate a variety of sensor geometries.

There are three stages to developing an application. First, we design a prototype program on the development host. At the first stage there are few memory, timing or numeric precision restrictions. The second stage involves debugging the application on the target hardware, under the more stringent hardware restrictions. Such restrictions are a result of the inherent tradeoff between a highly flexible, programmable development system and an inexpensive target system. In the third stage, we burn the application into the ROM of the target system at which point we can take the target system "into the field". At present, we have implemented some programs through stage three, for example simple motion tracking and connected components analysis, while others, like an attentional OCR program for reading license plates [13] are still at stage one. Since C language development environments are available for both DSP and microprocessors used in this system, we develop C code on the Sun host, then download it to the target system.

3.2 Sensor-independent image processing

The *Apply* language [23][25] generalized local image processing for arbitrary processor architectures. Using a structure called a *connectivity graph*, we generalized local image processing operations to arbitrary sensor geometries. The two together lead to a programming abstraction for local image processing which is independent of both sensor geometry and processing architecture.

The connectivity graph explicitly represents the neighbor relations between pixels, which may be irregular for space-variant sensor geometries. Using this graph, we define a variety of image processing operations. For example, if p represents a pixel, $L(p)$ its gray value, and $\mathcal{N}(p)$ its neighbors, then we can define a simple edge operator $e(p)$ as

$$e(p) = \frac{1}{|\mathcal{N}(p)|} \sum_{q \in \mathcal{N}(p)} (L(p) - L(q))^2. \tag{1}$$

Note that the definition of $e(p)$ contains no special cases for pixels having different number of neighbors, and thus applies equally to pixels at the boundary of the image and to those in the interior.

Another simple example is the Laplacian, in sensor coordinates, $\ell(p)$ given by

$$\ell(p) = |\mathcal{N}(p)| \ L(p) \ - \sum_{q \in \mathcal{N}(p)} L(q) \qquad (2)$$

Using such definitions we can build a library of image processing routines that is independent of the sensor geometry. We report the details of the connectivity graph in another paper [22], where we show how to develop programs for template matching, connected components analysis, pan-tilt calibration, and pyramid operations.

4 Conclusion and discussion

We have drawn two major conclusions from this work.

- Given the logarithmic structured sensor architecture, relatively little processing power is required to perform significant real-time vision. Our current system requires roughly 2 MIPS. For a future version, we estimate that 50 MIPS would provide enough power for extensive real-time applications of computer vision.
- Although we are currently building a custom VLSI sensor, we have been able to emulate a logmap sensor with commercially available hardware. This sensor architecture, coupled with currently available general purpose microcontrollers and DSP's, enables the building of a high performance machine vision system without extensive investment in custom hardware.

The long-range goal of this project is to demonstrate that major new applications of robotics will become feasible when small low-cost machine vision systems can be mass-produced. This notion of "commodity robotics" is expected to parallel the impact of the personal computer, in the sense of opening up new application niches for what has until now been expensive and therefore limited technology.

References

[1] A. L. Abbot and N. Ahuja "Active surface reconstruction by integrating focus, vergence, stereo, and camera calibration" in *Proceedings of 1991 International Conference on Computer Vision.*

[2] A. L. Abbot and N. Ahuja "The University of Illinois Active Vision System", University of Illinois Beckman Institute Working Paper CV-91-8-2.

[3] P. K. Allen, B. Yoshimi and A. Timcenko "Real-time visual servoing" in *Proceedings of 1991 IEEE International Conference on Robotics and Automation* Sacramento, CA, April, 1991.

[4] A. Baloch and A. M. Waxman "Behavioral control of the mobile robot Mavin" Chapter 6 in *Neural Networks: Concepts, Applications, and Implementations* Volume IV. Edgewood Cliffs: Prentice-Hall, 1991.

[5] B. Bederson, R. Wallace and E. Schwartz "Two pantilt mechanisms" submitted to IEEE Robotics and Automation conference, 1992.

[6] C. Brown, ed. "The Rochester Robot" Computer Science Dept., University of Rochester, TR 257, August, 1988.

[7] J. Clark, and N. Ferrier "Modal control of an attentive vision system," *Proceedings of the Second International Conference on Computer Vision*, pp. 514-523, 1988.

[8] J. Crisman and J. Webb "The Warp machine on Navlab" *IEEE Pattern Analysis and Machine Intelligence*, Vol 13. No. 5., May, 1991.

[9] E. Dickmanns and V. Graefe "Applications of Dynamic Monocular Machine Vision," *Machine Vision and Applications*, Vol. 1, pp. 241-261, 1988.

[10] H. Kawarabayashi, M. Watanabe, Y. Shirai, M. Asada, and J. Miura "Tracking of a moving object using an active vision system", in *Proceedings of Japan Mechanical Society Conference on Robotics and Mechatronics*, Vol. B., June, 1991 (in Japanese).

[11] E. Krotkov, F. Fuma, and J. Summers "An Agile Stereo Camera System for Flexible Image Acquisition," *IEEE Journal of Robotics and Automation*, Vol. 4, No. 1, pp. 108-113, Feb. 1988.

[12] J. Little, G. Blelloch and T. Cass "Algorithmic techniques for computer vision on a fine-grained parallel machine", *IEEE Pattern Analysis and Machine Intelligence*, Vol 11. No. 3., March, 1989.

[13] P.-W. Ong, R. Wallace and E. Schwartz "Space-variant optical character recognition", submitted to 1992 ICPR: pattern recognition systems and methodologies section.

[14] A. S. Rojer and E. L. Schwartz "Design considerations for a space-variant sensor with complex-logarithmic geometry" *10th International Conference on Pattern Recognition*, Atlantic City, NJ, pp. 278-85, 1990.

[15] A. S. Rojer *Space-variant computer vision with a complex-logarithmic sensor geometry*, Ph.D. Thesis, New York University, 1989.

[16] G. Sandini, F. Bosero, F. Bottino, and A. Ceccherini "The use of an anthropomorphic visual sensor for motion estimation and object tracking", in Proceedings of *Image Understanding and Machine Vision*, June, 1989.

[17] G. Sandini and P. Dario, "Active vision based on space-variant sensing" *International Symposium on Robotics Research* August, 1989.

[18] E. Schwartz "Spatial mapping in primate sensory projection" *Biological Cybernetics* vol. 25: pp. 181-194, 1977.

[19] E. Schwartz, B. Merker, E. Wolfson, and A. Shaw "Computational neuroscience: Applications of computer graphics and image processing to the two and three dimensional modeling of the functional architecture of visual cortex" *IEEE Computer Graphics and Applications* vol. 8, pp.13-28, 1988.

[20] M. Swain, ed. *Final Report of the Active Vision Workshop*, University of Chicago, August, 1991.

[21] J. van der Spiegel, G. Kreider, C. Claeys, I. Debusschere, G. Sandini, P. Dario, F. Fantini, P. Belluti, and G. Soncini "A foveated retina-like sensor using CCD technology" in *Analog Implementations of Neural Networks* eds. C. Mead and M. Ismail, Kluwer Publishing, 1989.

[22] R. Wallace, B. Bederson, P.-W. Ong, and E. Schwartz "Connectivity graphs for space-variant active vision", Vision Applications, Inc. Technical Report VAI-1, 1991.

[23] R. Wallace, J. Webb and I.-C. Wu "Architecture independent image processing: Performance of Apply on diverse architectures", in *Computer Vision, Graphics and Image Processing* vol. 48 pp. 265-276, 1989.

[24] R. Wallace and M. Howard "HBA Vision architecture: built and benchmarked", *IEEE Pattern Analysis and Machine Intelligence*, Vol 11. No. 3., March, 1989.

[25] J. Webb, L. Hamey and I.-C. Wu "An architecture independent programming language for low-level vision" *Computer Vision, Graphics and Image Processing* vol. 48, p. 246-264, 1989.

[26] C. Weiman and G. Chaikin "Logarithmic spiral grids for image processing and display" *Computer Graphics and Image Processing* vol. 11, pp. 197-226.