Chapter 5

Results

5.1 System architecture

Cortex-I consists of a camera, actuator, camera driver, motor driver, video interface, telephone interface, and processors. The three processors communicate via high-speed serial interface. The flow of information between the modules (Figure 5.1) starts with first DSP board that controls the camera driver, acquires the uniform sensor image, and maps it to a logmap image. It sends the image to the M68332-based microcontroller board which performs image processing, and controls the motor. The microcontroller board sends the resulting image to the second DSP board that maps the logmap image to the inverse map image and creates an RS-170 video signal for display on a TV monitor. The microcontroller board also sends the image to an optional third DSP board that is used as a video transmitter that can send 4 frames per second of logmap images over a standard voice-grade telephone line [56].[1]

Each of the processors 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 Sparcstation 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 an Analogic video digitizer, with software tools to simulate a variety of sensor geometries.

There are four stages to developing an application.

1. We design a prototype program on the development host. At the first stage there are few memory, timing or numeric precision restrictions. The commercial camera and actuator provide a high degree of reliability.

2. Development continues on the host, but we use the camera and actuator of **Cortex-I**. The host controls the SPM by sending commands over an RS-232 serial line, and receives the logmap image from the miniature camera, digitizing the video signal output. This allows the the same software to be tested using the new hardware.

3. The application is debugged on the target hardware using the host as a terminal. Since *C* language cross-development environments are available for both the DSPs and the microcontroller used in this system, we develop *C* code on the Sun host, then download it to the target system. The code must account for the more stringent hardware restrictions such as integer arithmetic, less memory, and less processing power. Such restrictions are a result of the inherent tradeoff between a highly flexible, programmable development system and an inexpensive target system.

[1] Patent application #07/802,320, entitled “Telephone Line Picture Transmission” was submitted to the U.S. Patent Office in 1991.

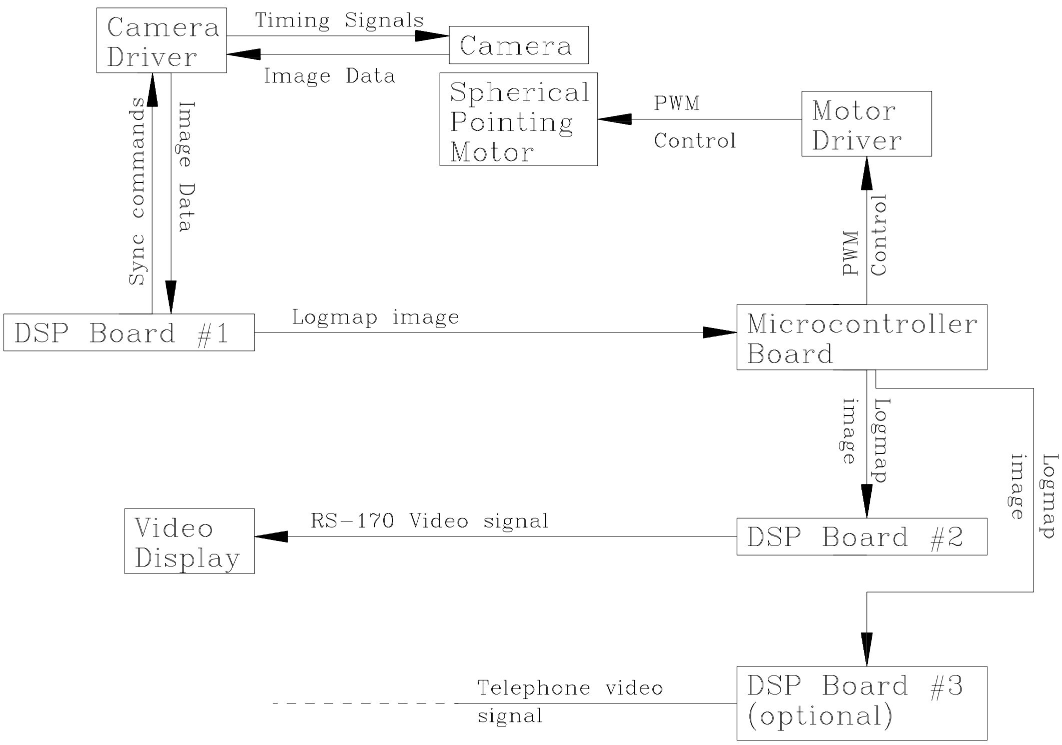

Figure 5.1: System architecture overview.

4. We burn the application into the ROM of the target system at which point we can take the target system "into the field". Some applications, like an attentional OCR program for reading license plates [38] are at stage two, while many simpler image processing algorithms, such as motion tracking are at stage four.

5.2 Image processing

We have developed several algorithms that are are burned into ROM in **Cortex-I**, and are transportable. some of them use the connectivity graph defined in Section 1.4 of Chapter 1.

Image binarization using the mean pixel value for a threshold runs at 22 frames per second (fps). The mean pixel value is given by

$$\overline{p} \leftarrow \frac{1}{\mathsf{NPIXELS}} \sum_{p \in L} L(p)$$

Then the binarized image, L_B, is defined by

$$L_B(p) \leftarrow \begin{cases} 0 & \text{if} L(p) \leq \overline{p} \\ 255 & \text{otherwise} \end{cases}$$

Image smoothing runs at 3 fps. The smoothed image, L_S, is defined by

$$L_S(p) \leftarrow \frac{1}{|\mathcal{N}(p)|} \sum_{q \in \mathcal{N}(p)} L(q)$$

We defined a simple edge operator that runs at 2 fps. The edge image, L_E, is defined by

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

Finally, a connected components algorithm runs at only 1 fps, illustrating both the generality of the programming model and the limitations of the microcontroller in the current prototype. The connected components problem begins with a given predicate F, for example the foreground-background function

$$F(p,q) = \begin{cases} \text{true} & \text{if} L_B(p) = L_B(q) \\ \text{false} & \text{otherwise} \end{cases} \tag{51}$$

$F(p,q)$ is true in a binary image when the two pixels p and q are either both black or both white. The algorithm to find all connected components of pixels satisfying F is well-known [7,31,44]. We use the familiar graph theoretic algorithm [19]. First we break the graph into subgraphs by removing those edges between pixels where $F(p,q)$ is false. Then, by the connectedness and components algorithm for graphs, our program extracts the connected regions. Figure 5.2 shows the result.

5.3 Motion tracking

We have implemented a motion-tracking application on **Cortex-I** in ROM that integrates all the capabilities of the system. It is based on a simple frame-differencing algorithm and is able to track moving objects at approximately 5 frames per second.

The algorithm, **motion_track**, subtracts successive pairs of frames, computes the centroid of motion, and moves the motor to point the camera at the computed centroid. Because this algorithm needs two frames to compute the centroid of motion, and then it misses a frame while the motor is moving, it can only make one motor movement every three frames. Therefore, although it can compute the frame difference and centroid at 16 Hz, the motor is only moved about 5 times per second.

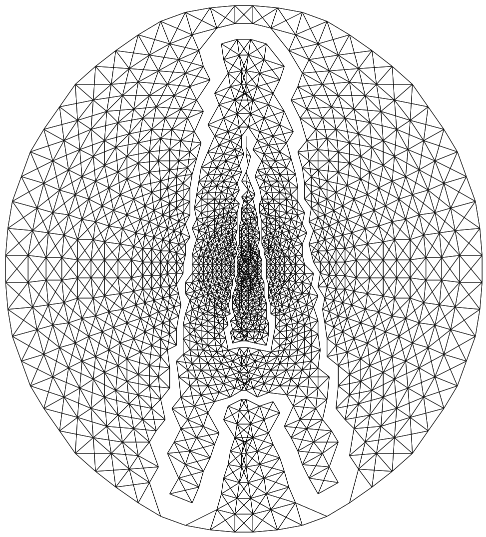

Figure 5.2: The connectivity graph reduces the image connected components problem to its graph theory counterpart. Here we show the foreground-background segmentation of an image containing the letter "A".

Algorithm: **motion_track**
Function: Moves the motor to keep the camera foveated on the centroid of motion in the scene.
Input: None
Output: None

$\Theta \leftarrow 0$ { Initial motor position }
$\Phi \leftarrow 0$
Repeat
  Move motor to (Θ, Φ)
  wait 50ms { Wait a little for motor ringing to decay }
  $L_{Null} \leftarrow$ Acquire image { Ignore image with motor movement }
  $L_1 \leftarrow$ Acquire image { Acquire two successive images }
  $L_2 \leftarrow$ Acquire image
  $\mathcal{P} \leftarrow 0$ { $\mathcal{P}$ in TV image coordinates }
  total_weight $\leftarrow$ 0
  For each point, $p \in L_1$ { Compute centroid of motion }
    $L_M(p) \leftarrow |L_1(p) - L_2(p)|$
    If $L_M(p) >$ MOTION_THRESH **Then**
      weight $\leftarrow L_M(p) \times a(p)$
      $\mathcal{P} \leftarrow \mathcal{P}$ + weight $\times\ \boldsymbol{\mu}(p)$
      total_weight $\leftarrow$ total_weight + weight
    End If
  End For
  $(x, y) \leftarrow \mathcal{P}$/total_weight
  $\Theta \leftarrow \Theta$ + DEG_PER_PIXEL $\times\ y$

$\Phi \leftarrow \Phi + \mathsf{DEG\_PER\_PIXEL} \times x$
Forever

5.4 License plate tracking & reading

We solved a difficult tracking and pattern recognition problem using a combination of **Cortex-I** and a Sun Sparcstation, bringing it to the second stage of development as discussed in Section 5.1. This application is described in detail in [37,38], but is summarized here. The task is to find the license plate on a moving car, track it as the car moves towards the camera, and finally read the characters on the license plate when it gets close enough to the camera.

The experimental setup consists of a real license plate attached to a toy truck that is pulled by a string wound around a motor-driven pulley. The truck moves at approximately two inches/second. It is tracked for about 6 feet before the license plate is read.

Cortex-I is set up to allow commands sent via an RS-232 serial port to control the SPM position. The commands specify the pan and tilt angles of the motor in tenths of a degree. **Cortex-I** outputs a standard RS-170 video signal that contains the inverse logmap image. This signal is then digitized into the Sun, and the forward logmap is computed from the inverse image just as if it were a uniform TV image. This method is inefficient as it wastes communication bandwidth in transmitting an inverse image, and computational resources in computing the forward map from the inverse map. We chose this approach because it allows the existing application running on the host to use the **Cortex-I** hardware without modification.

The license plate is tracked by using a combination of three techniques which are a modified hough transform [43], a foreground-background segmentation, and a corner detector. The algorithm assumes that the car moves at a constant velocity. It estimates the license plate's position each frame using a Kalman filter. When the car is close enough, and thus the characters on the license plate are big enough, the license plate is scanned from left to right with 5 frames. Each frame is analyzed to segment the characters and perform OCR. The results from the 5 frames are then merged together to yield the string of characters on the license plate. A sequence of images showing the tracking of the license plate is shown in Figure 5.3. The OCR results are shown in Figure 5.4, and the actual and estimated motor position during the tracking phase is shown in Figure 5.5.

This application also runs on the Sun development system using a commercial camera and actuator where it correctly reads the license plate [37]. The present prototype's limitations are described in the next section. We are currently building a new version which addresses these limitations - adding a zoom lens, more processing power, and more memory.

5.5 Cortex-I limitations

The development of the application described in the previous section allows for a better understanding of some of the limitations of using **Cortex-I** in a real situation. We found that although **Cortex-I** works well, it is limited in several ways which are:

Not enough processing power The Motorola M68332, which is capable of about 2 MIPS, performs all of the image processing in the system as well as controlling the motor. This is not enough to track the license plate even at these low speeds.

Not enough memory The M68332 board has only 192 Kbytes of RAM. This is not enough to hold the OCR tables, and thus the application could not be run on **Cortex-I** directly.

Not enough sensor resolution By the time the license plate is near enough to the camera so that the characters are reliably recognizable, the truck is so close to **Cortex-I**, that there is not enough time to scan the license plate and read the characters. Because of this, the characters on the

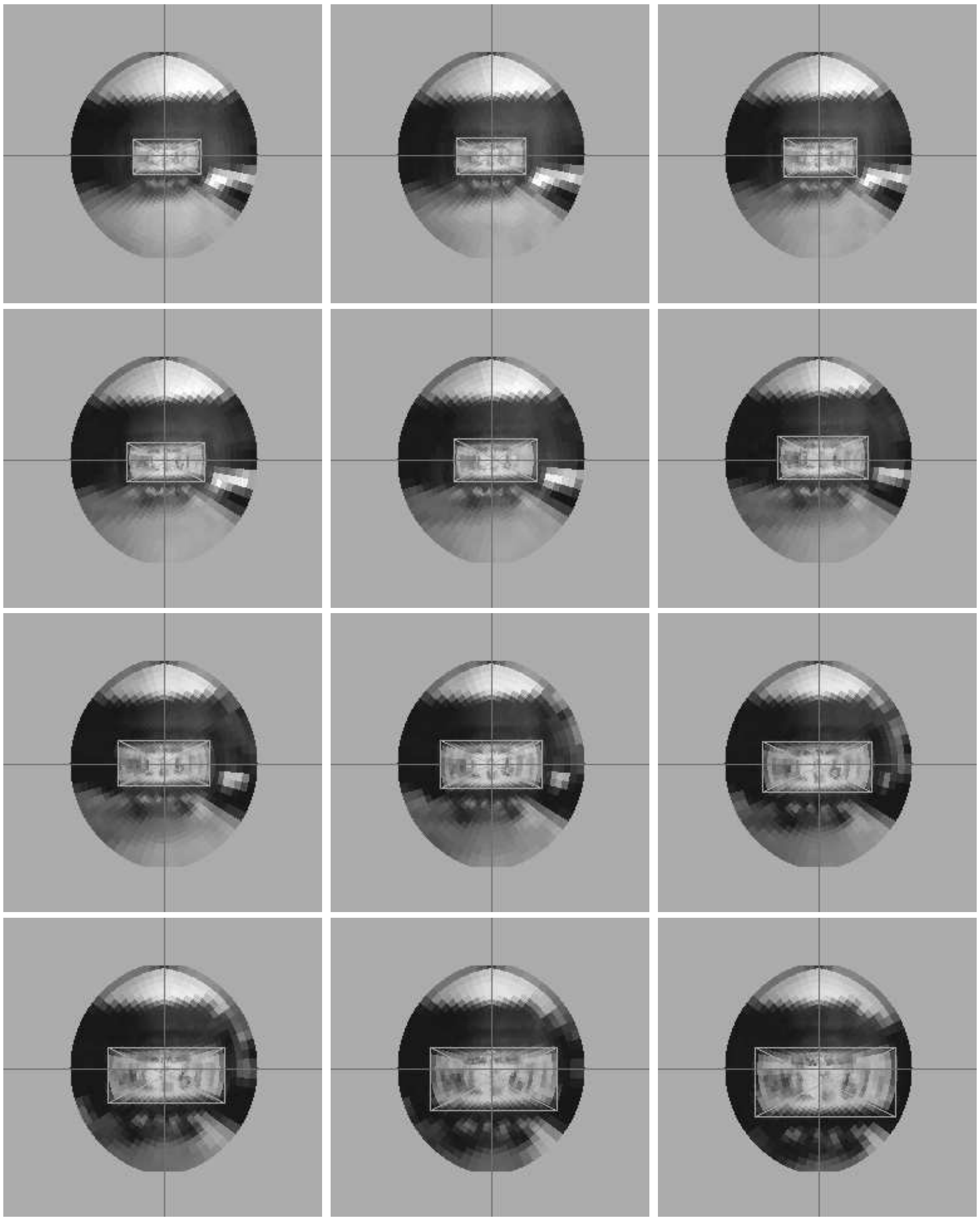

Figure 5.3: Sequence of images showing the results of tracking a moving license plate using a Kalman filter estimation procedure. A rectangle is drawn around the license plate showing that the license plate has been found correctly.

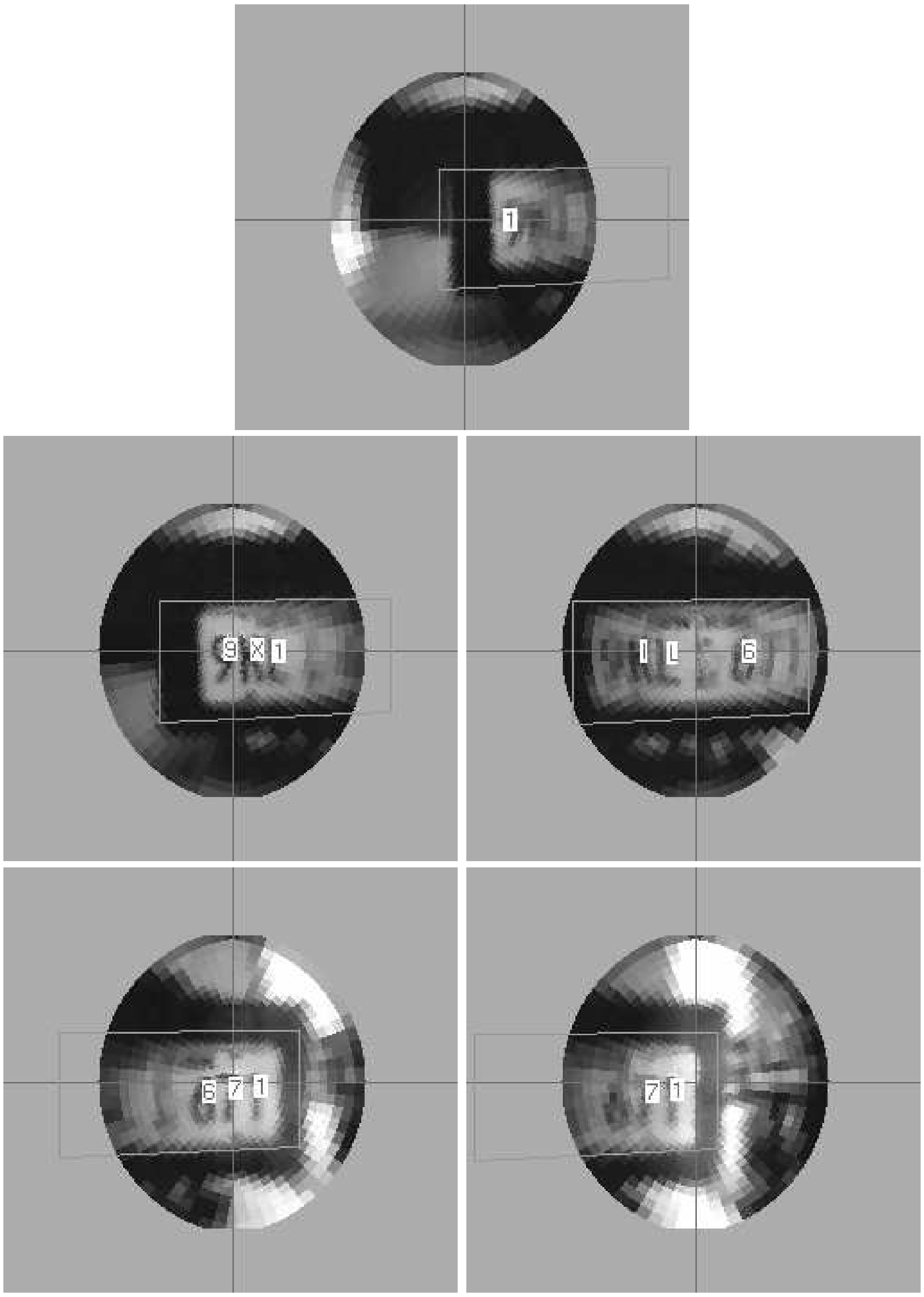

Figure 5.4: The OCR results obtained with several fixation points while the license plate was still moving. The OCR results are superimposed on the images. The actual text on the license plate is "9KL 671". The program output was "9X1L 6711".

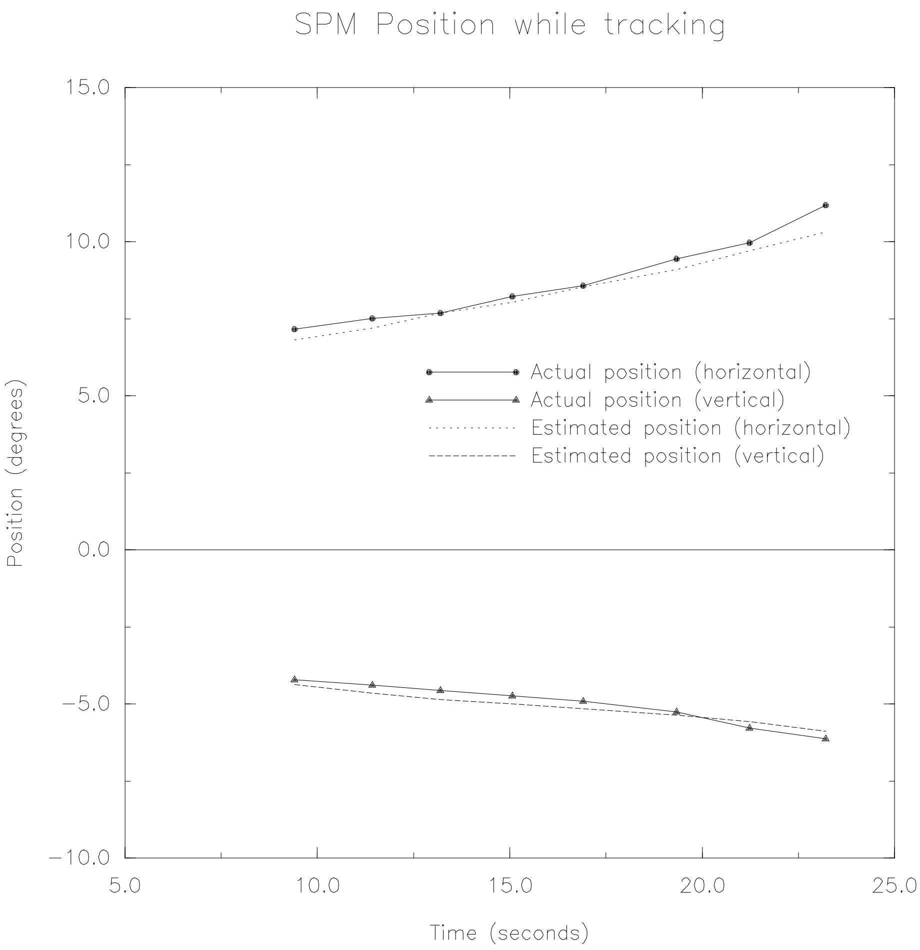

Figure 5.5: Horizontal and vertical actual and estimated position of the motor while tracking a license plate on the moving toy car.

license plate must be read while the license plate is still too far away, resulting in the imperfect recognition results.

Camera doesn't rotate about its focal point The application software assumes the camera rotates about its focal point when it scans the license plate to read the characters. Because the camera doesn't rotate about its focal point, the rotation results in an effective translation which throws off the expected position of the camera. Because of this, the OCR results from the different frames are not merged accurately, yielding an incorrect resultant string.

Appendix A

Glossary

active vision A computer vision system that purposely changes the sensor's parameters in response to the sensor's environment. The parameters that can be changed very often include, but are not limited to, position, focus, zoom, and vergence.

actuator A device to put an object into action or motion.

attention Where to look next. In a space-variant active vision system, only a small area of the scene (i.e., the fovea) is viewed in high resolution, so the problem of where to aim the camera is the problem of attention.

CCD A silicon based technology, the Charged Coupled Device (CCD) is used almost exclusively for image sensors.

CMOS A silicon based technology, the Complementary Metal Oxide Semiconductor (CMOS) circuit allows many types of electronic circuits to be built, including image sensors.

connectivity graph The graph representation of a space-variant image that allows general-purpose image processing on an image of any geometry. The graph has one vertex for each pixel in the image, and one edge between every pair of pixels that are adjacent to each other in the inverse map.

dual two-step control strategy A control strategy for moving the SPM. This applies the two-step method twice.

field width/field of view A description of how much of an external scene is viewed by a camera, typically measured in degrees.

fixed percentage control strategy A control strategy for moving the SPM. This moves the motor in small decreasing steps to the final position.

forward map image The computer representation of a space-variant image. Each pixel is stored in one word of memory and represents differing amounts of the external scene resulting in a warped image.

fovea The central, high-resolution portion of a logmap image.

home position The initial resting position of the Spherical Pointing Motor from which it can make pan or tilt excursions of limited extent.

inverse map image A space-variant image mapped to scene coordinates so that the image is no longer warped, and represents the external scene. Its main purpose is for a human to view a space-variant image.

logmap image An example of a space-variant image defined by a complex logarithm. It is typically composed of concentric rings of pixels, so that the center of the image has very small pixels and high resolution while the periphery has large pixels and thus, low resolution.

multi two-step control strategy A control strategy for moving the SPM. This applies the two-step method multiple times.

multi-resolution Any of several techniques of image representation and processing in which a scene is represented by an image or images of varying resolution. Both pyramids and space-variant images are examples of multi-resolution techniques.

pan Movement of a object (such as a camera) horizontally.

pan-tilt actuator A computer-controlled actuator designed to point an object such as a camera.

Platform Pantilt A small two degree-of-freedom pan-tilt actuator that is used to point a small camera. It consists of a platform connected to two linear stepper motors by several universal joints. It is run open-loop, and attains rotational velocities of up to about 100 degrees per second.

progressive video coding A method of image compression in which a image is represented by a sequence of logmap images foveated at different points in the scene. A high-resolution version of the scene is built up over time from the multiple images.

pyramid A multi-resolution method of image representation where an image is stored at several different resolutions. Typically the lowest level contains the full image, the next level contains the image at half-resolution, the next level contains the image a quarter-resolution, and so on until the top level contains a single pixel.

random map An example of a space-variant image defined by a stochastic process that places pixels randomly in the map with more pixels in the center than the periphery.

resolution A measure of the information content of an image. Higher resolution means more information.

spoke and ring lookup tables The look-up-tables used to map a TV image to a space-variant image. The tables contain the address that each TV pixel maps to. Because it is a many-to-one mapping, all TV pixels that are mapped to the same forward map pixel must be averaged together.

ringing The oscillatory motion that the Spherical Pointing Motor makes when moved to a new position without any control.

runs Typically used in the description of "run-length-encoding", a run is a sequence of pixels in a row of an image that have the same value. We use it to specify a sequence of pixels in a row of a TV image that map to the same logmap pixel.

saccade Very fast point to point motions of the human eye that are used to attend to a new area, not for tracking a moving object. "Saccade" is often applied to a computerized pan-tilt actuator to describe any point to point motion.

solenoid A coil of wire that when carrying current, acts like a bar magnet. Typically used with a permanent magnet to make a one degree-of-freedom linear actuator.

space-variant image An image whose resolution is spatially non-uniform, an example of which is a logmap image.

spatial quality measure, Q_s A figure of merit for an image sensor describing the ratio of field-width to resolution.

Spherical Pointing Motor A miniature two degree-of-freedom pan-tilt actuator that is used to point a miniature camera. It is a single direct drive motor that is run open-loop and attains rotational velocities of several hundred degrees per second.

temporal continuity The concept that in real scenes, things change slowly. Thus, in a rapid sequence of images of a real scene, there is not much change between successive frames, and the position of an object in one frame is a good prediction of the position of that object in the next frame.

tilt Movement of an object (such as a camera) vertically.

truncated pyramid A multi-resolution method of image representation based on a pyramid where only a fixed-size window of each pyramid is stored. The result is effectively a digital approximation of a logmap image. The bottom layer contains a high-resolution representation of a small field in the image. The next layer contains half the resolution of twice as much area, and so on. The truncated pyramid is essentially a window into a regular pyramid where the window represents the foveation point.

TV/uniform image A "standard" image consisting of a rectangular array of pixels that are all the same size and represent equal areas of scene space. Sometimes referred to as a space-invariant image.

two-step control strategy A control strategy for moving the SPM. This moves the motor halfway to the desired position and when the motor overshoots and reaches its zero-velocity point, it is held at this final position.

vertical smearing The resulting image degradation of certain CCD image sensors produced because the image is read from the sensor by shifting it vertically through the sensor. Thus, as the image is read out from the sensor, it accumulates light intensity information from every other pixel in the same column.

Appendix B

Variable List

$a(u, v)$	Area of a logmap pixel in TV pixels
α	Constant used to offset logmap calculation to replace the singularity at the center with a discontinuity along the vertical meridian
A	Enclosed area of a coil of the SPM
$\mathcal{A}$	Matrix of coefficients used to desmear a logmap image
b	Vectorized version of $L'(i, j)$ used for desmearing
β	Constant used in the equation for the ringing of the SPM
B	Permanent magnetic field in the SPM
c	Damping constant used in the equation for the ringing of the SPM
C	The magnitude of the exponential envelope describing the decay of the ringing of the SPM
calib	Calibration table for the SPM to map coil currents to motor position
d	Distance between the center of the universal joint connected to the end of the second drive shaft in the Platform Pantilt
D	Diameter of the wire in a coil of the SPM
e	Distance that the laser beam moved when measuring SPM accuracy
E	All of the edges in the connectivity graph
E_I	Edges in the connectivity graph from the inverse map
E_F	Edges in the connectivity graph from the forward map
f	Distance of SPM from wall when measuring accuracy
$G(V, E)$	Connectivity graph of a logmap image
h_1	Height of the first shaft in the Platform Pantilt
h_2	Height of the second shaft in the Platform Pantilt
$I(i, j)$	TV image
$I'(i, j)$	TV image as it read out from a CCD image sensor that has vertical smearing
$\boldsymbol{\iota}$	Current in a coil of the SPM
i, j	General purpose indices, most often used for TV images
k, l	General purpose indices
κ	Constant used in the equation for torque of the SPM
K	Constant equal to NA used in describing the SPM
$L(u, v)$	Logmap image
$L'(i, j)$	Logmap image as it read out from a CCD image sensor that has vertical smearing

m	Object distance in a lens system
m'	Image distance in a lens system
N	Number of turns of wire in a coil of the SPM
$\mathcal{N}(p)$	List of neighbors of a logmap pixel, p
Θ	Tilt (vertical) angle of the SPM
p	Individual vertex in the connectivity graph, i.e., point in a logmap image
$\overline{p}$	Mean pixel value in a logmap image
P	Power dissipated by a coil of the SPM
P_1, P_2	Points in retinal and log space, respectively, used to illustrate the logmap calculation
$\mathcal{P}$	Point in TV image space
$\phi(u, v)$	Vertex identified by its pixel coordinates
Φ	Pan (horizontal) angle of the SPM
r	Rotational inertia of the SPM
$\mathcal{R}$	Resistance of a coil of the SPM
$R(i, j)$	Ring look-up-table for computing $L(u, v)$ from $I(i, j)$
run_addr	Run address table used by **run_length_encode** to map a TV image to a logmap image.
run_len	Run-length table used by **run_length_encode** to map a TV image to a logmap image.
s	Distance between the center of the universal joint connected to the end of the first drive shaft in the Platform Pantilt
$S(i, j)$	Spoke look-up-table for computing $L(u, v)$ from $I(i, j)$
Ψ	Difference between the initial position and the destination position of the SPM
$\boldsymbol{\tau}$	Torque on the SPM due to a coil of wire
threshold	Threshold used for binarizing an image used in calibrating the SPM.
u, v	General purpose indices, most often used for logmap images
$u(p), v(p)$	Array coordinates in logmap image L of a point p
$\overline{\boldsymbol{\mu}}$	Magnetic dipole moment induced by a coil of wire in the SPM
$\boldsymbol{\mu}(p)$	Centroid of a logmap pixel, p, in TV image coordinates
V	Vertices in the connectivity graph
$\mathcal{V}$	Voltage across a coil of the SPM
ω	Frequency of the ringing of the SPM
$W(u, v, k, l)$	Weights used in computing logmap desmearing
$W_{TOT}(u, v)$	Total weights used in computing logmap desmearing
x, y	Coordinates of a point in log space for computing a logmap image
$\mathcal{X}$	Desmeared logmap image
z	Point in retinal space used to illustrate logmap calculation

Appendix C

Constant List

DEG_OF_WORKSPACE	Size of workspace of SPM in degrees
DEG_PER_PIXEL	Number of degrees that each pixel subtends for our image sensor and lens combination
DIM_CALIB	Number of calibration points per axis for the SPM
INT_SCAN	Integration to scan-time ratio for reading out the CCD image sensor
MOTION_THRESH	Threshold used by **motion_track** algorithm to determine when a motion pixel should contribute to the motion centroid.
NCOLS	Number of columns in a TV image, $I(i,j)$, i.e., $j \in \{0,\ldots,\mathsf{NCOLS}-1\}$
NPIXELS	Total number of pixels in a logmap image, $L(u,v)$ that map to TV pixels, i.e, the count of pixels, (u,v) such that $a(u,v) > 0$
NRINGS	Number of columns in a logmap image, $L(u,v)$, i.e., $v \in \{0,\ldots,\mathsf{NRINGS}-1\}$
NROWS	Number of rows in a TV image, $I(i,j)$, i.e., $i \in \{0,\ldots,\mathsf{NROWS}-1\}$
NRUNS	Number of runs in the mapping from TV to logmap coordinates.
NSPOKES	Number of rows in a logmap image, $L(u,v)$, i.e., $u \in \{0,\ldots,\mathsf{NSPOKES}-1\}$

Bibliography

[1] A. Lynn Abbott. Selective fixation control for machine vision: A survey. In *Proceedings of the IEEE International conference on systems, man, and cybernetics*, pages 1–6, October 1991.

[2] A. Lynn Abbott and Narendra Ahuja. Surface reconstruction by dynamic integration of focus, camera vergence, and stereo. *International Conference on Computer Vision*, pages 532–543, 1988.

[3] A. Lynn Abbott and Narendra Ahuja. The university of illinois active vision system. Technical Report CV-91-8-2, University of Illinois, Beckman Institute, 1991.

[4] Peter K. Allen, Billibon Yoshimi, and Aleksandar Timcenko. Real-time visual servoing. In *Proceedings of the 1991 IEEE International Conf. on Robotcs and Automation*, pages 851–856, April 1991.

[5] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *Intl. J. Computer Vision*, 2:333–356, 1988.

[6] R. Bajcsy. Active perception. *IEEE Proceedings*, 76(8):996–1005, 1988.

[7] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ, 1982.

[8] Aijza A. Baloch, Allen M. Waxman, Aijza A. Baloch, and Allen M. Waxman. Visual learning: adaptive expectations, and behavioural conditioning of the mobile robot mavin. *Neural Networks*, 4:271–302, 1991.

[9] Leonard Bergstein. General theory of optically compensated varifocal system. *Journal of the Optical Society of America*, 48(3):154–171, 1958.

[10] Leonard Bergstein and Lloyd Motz. Two-component optically compensated varifocal system. *Journal of the Optical Society of America*, 52(4):353–362, 1962.

[11] P. J. Burt. Algorithms and architectures for smart sensing. *Proc. DARPA Image Understanding Workshop*, pages 139–153, 1988.

[12] Peter. J. Burt. Attention mechanisms for vision in a dynamic world. *9th IEEE Intl. Conf. on Pattern Recognition*, pages 977–987, 1988.

[13] ed. C. Brown. The rochester robot. Technical Report 257, University of Rochester, 1988.

[14] A. Califano, R. Kjeldsen, and R. M. Bolle. Data and model driven foveation. *10th International Conference on Pattern Recognition, Vol. 1*, pages 1–7, 1990.

[15] G. M. Chaikin and C. F. R. Weiman. Image processing system. U.S. Patent No. 4,267,573, May 1981.

[16] J. J. Clark and N. J. Ferrier. Modal control of an attentive vision system. *Second Intl. Conf. Computer Vision*, page 514, 1988.

[17] James L. Crowley. *A Representation for Visual Information*. PhD thesis, Carnegie-Mellon University, November 1981.

[18] James L. Crowley and Amy L. Lowrie. Multiple resolution representation and matching of stereo scan lines, January 1985.

[19] N. Deo. *Graph Theory with Applications to Engineering and Computer Science.* Prentice-Hall, 1974.

[20] E. D. Dickmanns and V. Graefe. Applications of dynamic monocular machine vision. *Machine Vision and Applications*, 1:241–261, 1988.

[21] E. D. Dickmanns and V. Graefe. Dynamic monocular machine vision. *Machine Vision and Applications*, 1:223–240, 1988.

[22] E. F. Fichter. A stewart platform-based manipulator: General theory and practical construction. *The International Journal of Robotics Research*, 5(2):157–182, 1986.

[23] David Halliday and Robert Resnick. *Fundamentals of Physics, Second Edition, Extended Version.* John Wiley & Sons, New York, 1981.

[24] Vincent Hayward and Ronald Kurtz. Preliminary study of serial-parallel redundant manipulator. In *NASA Conference on Telerobotics*, 1989.

[25] Paul Horowitz and Winfield Hill. *The Art of Electronics, 2nd Edition.* Cambridge University Press, 1989.

[26] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *J. Physiol. (Lond.)*, 195:215–243, 1968.

[27] D. H. Hubel and T. N. Wiesel. Sequence regularity and geometry of orientation columns in the monkey striate cortex. *J. Comp. Neurol.*, 158:267–293, 1974.

[28] Francis A. Jenkins and Harvey E. White. *Fundamentals of Optics.* McGraw-Hill, New York, 1957.

[29] Hideharu Kawarabayashi, Makoto Watanabe, Yoshiaki Shirai, Minoru Asade, and Jun Miura. Tracking of a moving object using an active vision system. *Japan Mechanical Society, Robotics Mechatronics Conference Proceedings*, pages 207–212, June 1991.

[30] T. Kenjo and S. Nagemori. *Permanent-Magnet and Brushless DC Motors.* Oxford Science Publications, 1985.

[31] Berthold Klaus and Paul Horn. *Robot Vision.* The MIT Press, Massachusetts, 1986.

[32] G. Kreider, J. Van der Spiegel, I. Born, C. Claeys, I. Debusschere, G. Sandini, P. Dario, and F. Fantini. A retina like space variant ccd sensor. *SPIE/Charge-Coupled Devices and Solid State Optical Sensors*, 1242, 1990.

[33] E. Krotkov, F. Fuma, and J. Summers. An agile stereo camera system for flexible image acquisition. *IEEE J. Robotics and Automation*, 4:108–113, 1988.

[34] Larry Matthies, Richard Szeliski, and Takeo Kanade. Incremental estimation of dense depth maps from image sequences. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, pages 366–374, 1986.

[35] C. Mead. *Analog VLSI and Neural Systems.* Addison-Wesley, 1989.

[36] Thomas J. Olson and Robert D. Potter. Real-time vergence control. Technical Report 264, University of Rochester, 1988.

[37] Ping-Wen Ong. *Image Processing, Pattern Recognition and Attentional Algorithms in a Space-Variant Active Vision System.* PhD thesis, GSAS, New York University, 1992, expected.

[38] Ping-Wen Ong, Richard S. Wallace, and Eric L. Schwartz. Space-variant optical character recognition. In *11th International Conference on Pattern Recognition*, 1992.

[39] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, NY, 1988.

[40] Daniel Raviv and Avraham Shapira. Miniature vision-based flight simulator. In *4th Annual Conference on Recent Advances in Robotics*, pages 81–86, 1991.

[41] David A. Robinson. The oculomotor control system: A review. *Proceedings of the IEEE*, 56(6):1032–1049, 1968.

[42] A. S. Rojer and E. L. Schwartz. Design considerations for a space-variant visual sensor with complex-logarithmic geometry. *10th ICPR, Vol. 2*, pages 278–285, 1990.

[43] Alan S. Rojer. *Space-variant computer vision with a complex-logarithmic sensor geometry.* PhD thesis, GSAS, New York University, 1989.

[44] Azriel Rosenfeld and Avinash C. Kak. *Digital Picture Processing, 2nd Edition.* Academic Press, 1982.

[45] G. Sandini, F. Bosero, F. Bottino, and A. Ceccherini. The use of an anthropomorphic visual sensor for motion estimation and object tracking. In *Image Understanding and Machine Vision Workshop*, pages 1–5, 1990.

[46] Giulio Sandini, Paolo Dario, and I. Debusschere. Active vision based on space-variant sensing. In *5th International Symposium on Robotics Research*, August 1989.

[47] E. L. Schwartz. The development of specific visual projections in the monkey and the goldfish: Outline of a geometric theory of receptotopic structure. *J. Theoretical Biology*, 69:655–685, 1977.

[48] E. L. Schwartz. Spatial mapping in primate sensory projection: analytic structure and relevance to perception. *Biological Cybernetics*, 25:181–194, 1977.

[49] E. L. Schwartz, B. Merker, E. Wolfson, and A. Shaw. Computational neuroscience: Applications of computer graphics and image processing to two and three dimensional modeling of the functional architecture of visual cortex. *IEEE Computer Graphics and Applications*, 8(4):13–28 (July), 1988.

[50] Neil C. Singer and Warren P. Seering. An extension of command shaping methods for controlling residual vibration using frequency sampling. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 800–805, Nice, France, May 1992.

[51] Warren J. Smith. *Modern Optical Engineering.* McGraw-Hill, New York, 1990.

[52] N. Stelman. Design and control of a six-degree-of-freedom platform with variable admittance. Master's thesis, Massachusetts Institute of Technology, Dept. of Mechanical Engineering, 1988.

[53] D. Stewart. A platform with six degrees of freedom. *Proceedings of the Institution of Mechanical Engineers*, 1965.

[54] John K. Tsotsos. On the relative complexity of active vs. passive visual search. *International journal of computer vision*, 7(2):127–141, 1992.

[55] J. van der Spiegel, F. 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 C. Mead and M. Ismail, editors, *Analog VLSI Implementations of Neural Networks.* Kluwer, Boston, 1989.

[56] Richard S. Wallace, Benjamin B. Bederson, and Eric L. Schwartz. Telecortex-i logmap video telephone. In *Proceedings of the IEEE Workshop on Applications of Computer Vision (submitted)*, November 1992.

[57] Richard S. Wallace, Ping-Wen Ong, Benjamin B. Bederson, and Eric L. Schwartz. Space-variant image processing. Technical Report 256, New York University, Computer Science, Robotics Research, October 1991.

[58] A. L. Yarbus. *Eye Movements and Vision.* Plenum, NY, 1967.

[59] Y. Yeshurun and E. L. Schwartz. Cepstral filtering on a columnar image architecture: a fast algorithm for binocular stereo segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(7):759–767, 1989.