

INTELLIGENT ROBOT VISION SENSORS IN VLSI

Ralph Etienne-Cummings

Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD 21218

etienne@ece.jhu.edu

Abstract

Traditional approaches for solving real-world problems using computer vision have depended heavily on CCD cameras and workstations. As the computation power of workstations doubles every 1.5 years, they are now better able to handle the large amount of data presented by the cameras; yet real-time solutions for physical interaction with the real-world continues to be very hard, and relegated to large and expensive systems. Our approach attempts to solve this problem by using computational sensors and small/inexpensive embedded processors. The computational sensors are custom designed to reduce the amount of data collected, to extract only relevant information and to present this information to the simple processor, microcontrollers (μ Cs) or DSPs, in a format which minimizes post-processing latency. Consequently, the post-processors are required to perform only high level computation on features rather than data. These systems are applied to problems such as target acquisition and tracking for image stabilization and autonomous data driven autonavigation for mobile robots. We present an example a system that uses a pair of computational sensors and a μ C to solve a toy autonavigation problem.

The computational sensors, however, have wide applications in many problems that require image preprocessing such as edge detection, motion detection, centroid localization and other spatiotemporal processing. This paper also presents a general-purpose computational sensor capable of extracting many visual information components at the focal plane.

I. INTRODUCTION

Consider a hostage situation in an urban environment. When the law enforcement individuals arrive on the scene, it would not be prudent to enter the building containing armed terrorists. Instead a group of semi-autonomous robots are placed at the doorway and they search the building to find the captives. The robots are given a variety of visual

sensors. They range from passive CCD cameras, IR sensors to intelligent sensors. The CCD and IR cameras are used to simply image the scene and allow the law enforcement officers to inspect the environment. Hence, they are off-the-shelf systems that can be easily interfaced with transmission hardware for viewing on PDAs (personal digital assistants). The computational sensors, on the other hand, are used to preprocess visual images; the extracted visual information is used to guide the robots. The robots autonomously execute the officers' command using the information provided by the computational sensors and decisions made by local/remote processing hardware.

To ensure their survival, the robots must have a variety of specialty skills, such as stealth (quiet, low EM signature), hazard detection (obstacle avoidance and evasion), speech recognition and homing, and cooperative map building (landmark recognition and location), as shown in figure 1. Clearly these are very complex and computationally intensive behaviors and properties. The combination of computational sensors, imagers, local/remote processors and clever algorithms will be instrumental in realizing these behaviors.

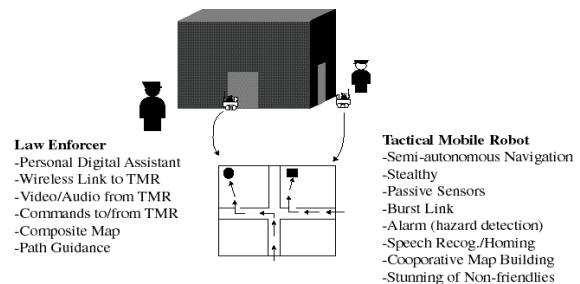


Figure 1: The Semi-Autonomous Urban Search/Locate Robot

Over the past few years, a few information extracting computational sensors have been developed [Koch, 1995]. The application of these sensors to real problems requiring visually guided interaction with the environment is still in its infancy [Kramer, 1998]. Consequently, the problems that have been attempted are small and relatively easy for the robotics community. They are, however, the sentry problems that evaluate the potential of using

computational sensors in difficult and complex robotics systems and tasks [Etienne, 1998]. So far, the results have been encouraging. We are now poised to expand the computational power of these smart sensors such that they can be used in real robot vision systems. Yet, as the speed and computational power of $\mu\text{P}/\text{DSP}/\mu\text{C}$ continues to increase while their cost plummets, can custom designed smart sensors compete with the ubiquitous and programmable CCD/CPU robot vision systems?

Fundamentally, having a computational sensor in the processing path is beneficial. This is because it relieves the $\mu\text{P}/\text{DSP}/\mu\text{C}$ from menial tasks of data preprocessing for information/feature extraction. Consequently, the $\mu\text{P}/\text{DSP}/\mu\text{C}$ can be applied to more difficult problems that capitalize on their general-purpose computational strengths. Despite the obvious power, speed and efficiency advantages of application specific computational sensors, the man-time and cost associated with their re-design for each new application are prohibiting. Hence, I propose a general-purpose computation sensor (GPCS) that performs considerable and programmable computation at the focal plane. The user has the advantage of choosing which processed image format and which process parameters are required for the problem. Furthermore, the same sensor can be used in a variety of robotics system, resulting in compatibility across systems and problems. The GPCS can subsequently replace the camera. Similar to other electronics commodities, such as CCDs, CPUs and DSPs, an industry can be constructed around providing these smart sensors with various specialties. Clearly, their interface and programming architecture must be standardized for general use.

As can be expected, the GPCS has a considerable design penalty, in terms of man-time and material cost. Consequently, these following questions beg answers.

- Should sensor designers focus on application-specific or general-purpose computational sensors?
- Is the prototyping cost of ASCS out-weighed by their performance benefit?
- Does flexibility of the GPCS out-weighs its redundancies and inefficiencies?
- Will the development of advanced computation sensors advance the state-of-the-art of robotics?

These questions are addressed below.

The examples of computational sensors for robot vision provided below addresses these two extremes.

The first design uses an application specific approach to realize a sensor with centroid localization properties [Etienne, 1998]. This chip is optimized for area compactness, low power and high speed. To demonstrate the usefulness of this computational sensor, it is combined with a simple μC to realize a data driven autonomously navigating mobile robot. The complete system is low-cost, compact, light-weight and operates at speeds up to 2 m/s. Power consumption is dominated by locomotion and not information processing. On the other hand, to overcome the shortcomings of the application specific approach, a general-purpose computational sensor (GPCS) for visual information processing is proposed. Performing ten complex information (feature) extraction tasks at the focal plane, this sensor is clearly a complicated IC. Consequently, it requires a great deal of design time. Yet, it offers a unique opportunity to standardize the front-end of all robot vision systems. Arguments are offered which highlight the benefits of using the GPCS with a $\mu\text{C}/\text{DSP}/\mu\text{P}$ processing core to implement robot vision systems.

II. MOTION CENTROID LOCALIZATION COMPUTATIONAL SENSOR

A. System Overview

Motion centroid computation is used to isolate the location of moving targets on a 2D focal plane array. Using the centroid computation, a chip which realizes a neuromorphic visual target acquisition system based on the saccadic generation mechanism of primates can be implemented. The biological saccade generation process is mediated by the superior colliculus, which contains a map of the visual field [Sparks, 1990]. Horiuchi has built an analog system that replicates most of the neural circuits (including the motor system) believed to form the saccadic system [Horiuchi, 1996]. Staying true to the anatomy forced his implementation to be a complex multi-chip system with many control parameters. On the other hand, our approach focuses on realizing a compact single chip solution by only mimicking the behavior of the saccadic system, but not its structure. The benefit of our approach is its compactness, low-power consumption and large response dynamic range.

B. Hardware Implementation

Our approach uses a combination of analog and digital circuits to implement the functions of the retina and superior colliculus at the focal plane. The

retina portion of this chip uses photodiodes, logarithmic compression, edge detection and zero-crossing circuits. These circuits mimic the first three layers of cells in the retina with mixed sub-threshold and strong inversion circuits. The edge detection circuit is realized with an approximation of the Laplacian operator implemented using the difference between a smooth (with a resistive grid) and original versions of the image [Mead, 1989]. The high gain of the difference circuit creates a binary image of approximate zero-crossings. After this point, the computation is performed using mixed analog/digital circuits. The zero-crossings are fed to ON-set detectors (positive temporal derivatives) which signal the location of moving or flashing targets. These circuits model the behavior of some of the amacrine and ganglion cells of the primate retina [Barlow, 1982]. These first layers of processing constitute all the “direct” mimicry of the biological models. Figure 2 shows the schematic of these early processing layers.

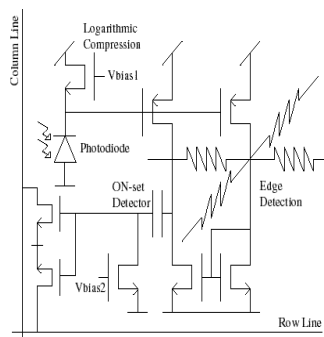


Figure 2: Schematic of the model of the retina.

The ON-set detectors provide inputs to the model of the superior colliculus circuits [Sparks, 1990]. The ON-set detectors allow us to segment moving targets against textured backgrounds. This is an improvement on earlier centroid and saccade chips which used pixel intensity [DeWeerth, 1992]. The essence of the superior colliculus map is to locate the target to be foveated. In our case, the target chosen to be foveated will be moving. Here motion is define simply as the change in contrast over time. Motion, in this sense, is the earliest measurable attribute of the target that can trigger a saccade without requiring any high-level decision making. Subsequently, the coordinates of the location of motion must be extracted and provided to the motor drivers.

The circuits for locating the target are implemented entirely with mixed signal circuits. The ON-set detector is triggered when an edge of the target appears

at a pixel. At this time, the pixel broadcasts its location to the edge of the array by activating a row and column line. This row (column) signal sets a latch at the right (top) of the array. The latches asynchronously activate switches and the centroid of the activated positions is provided. The latches remain set until they are cleared by an external control signal. This control signal provides a time-window over which the centroid output is integrated. This has the effect of reducing noise by combining the outputs of pixels which are activated at different instances even if they are triggered by the same motion (an artifact of small fill factor focal plane image processing). Furthermore, the latches can be masked from the pixels’ output with a second control signal. This signal is used to de-activate the centroid circuit during a saccade (saccadic suppression). Figure 3 shows a portion of the schematic of the superior colliculus.

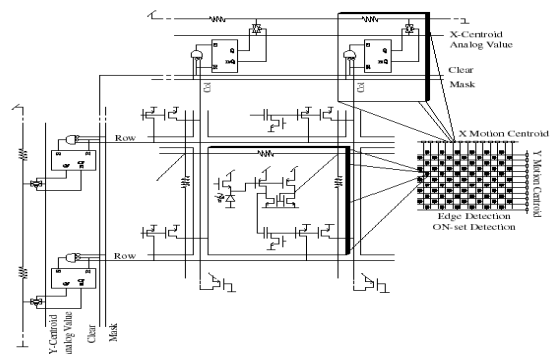


Figure 3: Schematic of the model of the superior colliculus.

C. Results

In contrast to previous work, this chip provides the 2D coordinates of the centroid of a moving target. Figure 4 shows the oscilloscope trace of the coordinates as a target moves back and forth (at a fixed y- displacement), in and out of the chip’s field of view. The y-coordinate does not changes while the x-coordinate increases and decreases as the target

Technology	1.2um ORBIT
Chip Size	4 mm ²
Array Size	17 x 19
Pixel Size	70 x 65 um ²
Fill Factor	11%
Intensity	0.1u - 100m W/cm ²
Min. Contrast	10%
Response Time	2 - 10 ⁶ Hz (@ 1 m W/cm ²)
Power (chip)	5 mW (@ 1 m W/cm ² , Vdd = 6V)

Table I: Performance characteristics of the centroid localization computational sensor.

moves to the left and right, respectively. The chip has been used to track targets in 2D by making micro-saccades. In this case, the chip chases the target as it attempts to escape from the center. The eye movement is performed by converting the analog coordinates into PWM signals, that are used to drive stepper motors. The system performance is limited by the contrast sensitivity of the edge detection circuit, and the frequency response of the edge (high frequency cut-off) and ON-set (low frequency cut-off) detectors. With the appropriate optics, it can track walking or running persons under indoor or outdoor lighting conditions at close or far distances. Table I summarizes the characteristics of the chip.

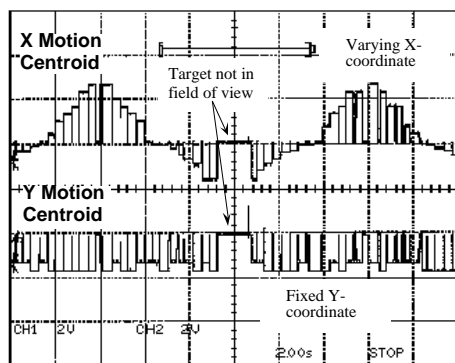


Figure 4: Oscilloscope trace of 2D centroid for a moving target.

III. AUTONOMOUS NAVIGATION USING THE COMPUTATIONAL SENSOR

A. System Overview

The simplest form of data driven auto-navigation is the line-following task. In this task, the robot must maintain a certain relationship with some visual cues that guide its motion. In the case of the line-follower, the visual system provides data regarding the state of the line relative to the vehicle, which results in controlling steering and/or speed. If obstacle avoidance is also required, auto-navigation is considerably more difficult. Our system handles line-following and obstacle avoidance by using two sensors which provide information to a micro-controller (μC). The μC steers, accelerates or decelerates the vehicle. The sensors, which use the centroid location system outlined above, provide information on the position of the line and obstacles to the μC . The μC provides PWM signals to the servos for controlling the vehicle. The algorithm implemented in the μC places the two sensors in competition with each other to force the line into a blind zone between the sensors. Simultaneously, if an object enters the visual field from outside, it is treated

as an obstacle and the μC turns the car away from the object. Obstacle avoidance is given higher priority than line-following to avoid collisions. The μC also keeps track of the direction of avoidance such that the

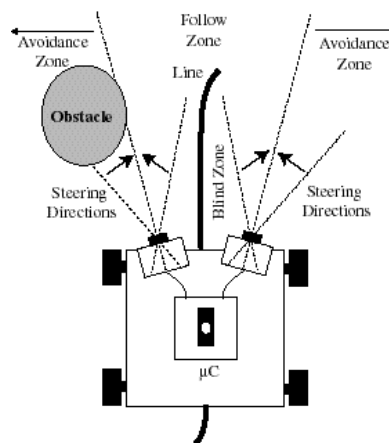


Figure 5: Block diagram of the autonomous line-follower system.

vehicle can be re-oriented towards the line after the obstacle is pushed out of the field of view. Lastly, for line following, the position, attitude and velocity of drift, determined from the temporal derivative of the centroid, are also used. The speed control strategy is to keep the line in the blind zone, while slowing down at corners, speeding up on straight aways and avoiding obstacles. The angle which the line or obstacle form with the x-axis also affects the speed. The value of the x-centroid relative to the y-centroid provides rudimentary estimate of the attitude of the line or obstacle to the vehicle. For example, angles less (greater) than ± 45 degrees tend to have small (large) x-coordinates and large (small) y-coordinates and require deceleration (acceleration). Figure 5 shows the organization of the sensors on the vehicle and control spatial zones. Figure 6 shows the vehicle and samples of the line and obstacles.

B. Hardware Implementation

The coordinates from the centroid localization circuits are presented to the μC for analysis. The μC used is the Microchip PIC16C74. This chip is chosen because of its five A/D inputs and three PWM outputs. The analog coordinates are presented directly to the A/D inputs. Two of the PWM outputs are connected to the steering and speed control servos. The PIC16C74 runs at 20 MHz and has 35 instructions, 4K by 8-b ROM and 80 by 20-b RAM. The PIC program determines the control action based on the signal provided by the sensors. The vehicle used is a four-wheel drive radio controlled model car

(the radio receiver is disconnected) with Digital Proportional Steering (DPS).

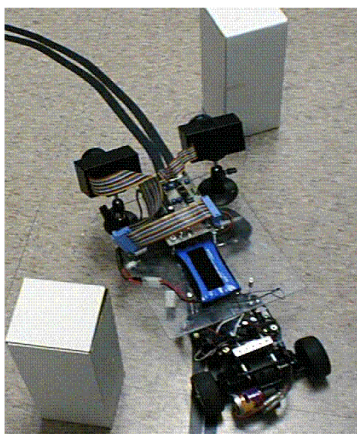


Figure 6: A picture of the vehicle.

C. Results

The vehicle was tested on a track composed of black tape on a gray linoleum floor with white obstacles. The track formed a closed loop with two sharp turns and some smooth S-curves. The sensors were equipped with 8mm variable iris lens, which limited their field of view to about 9° . Despite the narrow field of view, the car was able to navigate the track at an average speed of 1 m/s without making any errors. On less curvy parts of the track, it accelerated to about 2 m/s and slowed down at the corners. When the speed of the vehicle is scaled up, the errors made are mainly due to over steering and spinning-out in corners.

IV. A GENERAL PURPOSE VISUAL COMPUTATIONAL SENSOR

A. System Overview

The proposed general-purpose computational sensor (GPCS) is designed to offer a variety of programmable preprocessed images to higher level processing systems. The programmability offers flexibility to adapt the characteristics of the sensor to the environment, while not restricting its computational repertoire. We proposed a visual sensor which will (1) capture a 128×128 gray scale image using a CMOS imager core which can be fully scanned at thousands of frames/second, (2) offer logarithmic gain control for operation in a wide variety of ambient light intensity, (3) offer A/D

conversion on chip to simplify interface with DSP/CPU systems later in the computational pathway, (4) allow random access, region based and raster scanning output modes of the pixels, (5) offer programmable spatial processing (convolution engine using pipe-lined mixed signal circuitry) on the read-out, outputting many processed versions of the image in parallel, (6) offer temporal processing capabilities using local analog memories within each pixel, (7) perform spatiotemporal processing for movement detection and optical flow measurement, which will be output at frame rate, (8) output pixel, region and global scale motion vectors, (9) determine object centroid on a grid basis (programmable size) for multi-target tracking (covert and overt) and (10) perform all this computation in parallel with sub-millisecond latencies. Figure 7 shows the architecture of the sensor.

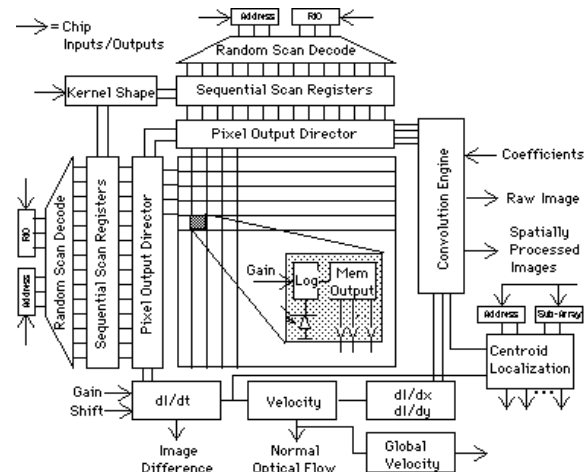


Figure 7: Architecture of the General Purpose Visual Computational Sensor.

The development of this GPSC is ongoing, but various core components have been constructed as stand-alone parts. Currently, items 1, 2, 5, 6 and 7 have been prototyped and tested. It will take at least another year to prototype the rest of components. A complete system is expected within a two-year time frame.

B. Fast 128×128 CMOS Imager

At the front of any vision system are phototransduction elements. In this sensor photodiodes are used due to their fast response characteristics [Sze, 1981]. This imaging system is implemented in standard VLSI technologies and CMOS circuits are used for focal plane image processing. The receptors perform continuous-time

current domain imaging because this approach allows direct processing of the image data. An additional benefit of current domain imaging is its fast read-out potential. Since the photocurrent is easily amplified within the pixel, relatively large currents are switched to the output circuitry. Furthermore, using node summing, spatial smoothing is easily implemented. Moreover, multiple copies of the pixel current are also obtained with one or two additional transistors. Lastly, logarithmic gain control is also readily available through the physics of transistor behavior.

C. Pixel Gain Control

For an imaging system to operate in a wide range of ambient lighting conditions, mechanical (pupil diameter) and/or electronic (integration time or pre-amplification gain) sensitivity controls are required to prevent output saturation in bright light. Electronic gain controls are much less expensive and simpler than their mechanical counterparts, so it is used in the computational sensor. Biological vision has developed gain control mechanism to allow operation over 10 orders of magnitude of ambient light [Normann, 1974]. CCD cameras exhibit severe limitations in this area. Their gain control mechanism usually respond with long latencies and a single control parameter is used for the entire array. Local gain control is more desirable for simultaneous viewing of bright and dark regions of images. The local gain control mechanism using a logarithmic circuit can be further augmented by a region based bias control strategy to further adapt the response of the pixels. This is easily done by controlling the bias voltage applied to the logarithmic transistor. We envision a control process mediated by an external processor, which can individually control the bias voltage applied to a particular pixel. By coordinating the scanning procedure with the bias control, the secondary local gain control mechanism can be implemented.

D. Analog-to-Digital Conversion

Incorporating ADC capabilities with imagers have become common practice over the past few years [Mendis, 1995]. This is due to the fact that CMOS imager can be easily integrated on the same chip with traditional analog and digital circuits. Since the GPCS is also implemented in CMOS technology, this capability will also be exploited. 8-bit ADCs will be integrated on the chip.

E. Pixel Access Strategy

The pixel access procedure is one of the most complicated components of this sensor. It must allow pixel and region addressing with random access or sequential scans. In addition, the scanning circuit also participates in the construction of the spatial and the temporal convolution kernels. The scanning modes will be partitioned into sequential and random scans, which will be selectable by external control logic. For either of these scan modes, the size of the region of interest is also programmable. In the sequential scan, the region of interest is the spatial scale over which convolutions are performed. In the case of the random scan scenario, two region sizes are defined. The first is a broad region defining the entire region of interest, while the second defines the size of the convolution kernels applied to the broad region of interest. Within the broad region of interest, the kernel is applied sequentially.

F. Spatial Processing

The realization of the spatial processing in the GPCS is heavily predicated on the availability of the desired data from the pixel. That is, the pixel outputs and the scanning circuitry must work together to provide the convolution kernels with the appropriately labeled signals [Etienne, 1998b]. For example, suppose that the image must be filtered with two Gabor filters oriented in the x and y direction, respectively. Figure 8 shows the two kernels. In order to implement these kernels, the central, first and second sub-spaces of the kernel must be identified and presented to the current scaling circuitry. This is realized by directing the pixel currents to the appropriate summing lines. Loading the central, first and second sub-spaces registers with the required bit patterns, the currents can be easily directed. The labeling is performed in both x and y so that both kernels are computed simultaneously. In addition, a smoothed version of the image is also computable by summing the currents from all or some of the sub-spaces. Furthermore, additional kernels can be realized by scaling and summing all the currents. The coefficients of the sub-space currents are also programmable and will support large kernels (up to 11 x 11 or 15 x 15, depending on need). Clearly, the kernels will not be completely general since all the currents in each sub-space is scaled by the same coefficient. This is a trade-off point, where the more flexibility translates into more wires in the imaging array (low imager density), more scanning registers and more current selection circuitry. With 3 sub-spaces, a wide variety

of kernels are possible and diagonal filters can be realized with some trickery.

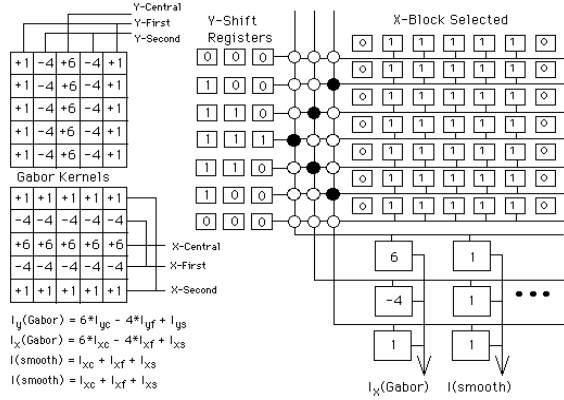


Figure 8: Evaluating spatial convolution in the x direction.

G. Temporal Processing

The temporal processing computation requires a delayed version of the image. A local analog memory, implemented as a sample-and-hold capacitor or an RC time constant, is placed within each pixel. The delayed image is sampled and held at the end of the read-out cycle of each pixel. There are two competing factors which must be balanced in designing the analog memory. Large sampling capacitors hold the pixel data for long periods, but the pixel will also be larger. By scanning the array faster, the analog memory will degrade less and faster changing intensities will be detected. The delayed image can be subtracted from the raw image to implement a temporal derivative that will be used for movement and velocity detection. In addition, to correct for errors in image stabilization during navigation, the delayed image can be easily shifted in x and y by programming the sensor. That is, usually the temporal derivative is given by $dI/dt = I(x,y,t) - I(x,y,t-1)$; the corrected temporal derivative will be $dI/dt = I(x,y,t) - I(x-m,y-n,t-1)$.

H. Movement and Velocity Detection

Movement detection is simply the difference between the raw image and the stored image. For velocity detection, the gradient based approach will be used [Horn, 1986]. In this paradigm, normal velocity is given by the ratio of the temporal derivative and the spatial derivatives of the image. That is, $v_x = (dI/dt)/(dI/dx)$ and $v_y = (dI/dt)/(dI/dy)$. To decrease noise, the spatial derivatives include smoothing orthogonal to the axis. This has the advantage of sampling the velocity vectors planes only in the x-

t and y- t planes, respectively, resulting in more accurate components of the normal optical field [Etienne, 1997]. Furthermore, the ratio is only computed if the confidence (i.e. the value) of the spatial derivative is high. The clear benefit of this step is to eliminate erroneous measurements in low contrast scenes, where the ratio will be singular.

I. Spatial Scale of the Motion Vectors

The spatial scale of the velocity field is controlled locally by the size of the spatial convolution kernel, since it determines the size of the image patch over which smoothing is performed. As the size of the kernel increases, more pixels are recruited into the spatial and temporal derivatives. Hence, the resulting vector field will be smooth, but local. For global velocity, an average must be computed over the entire array.

J. Object Centroid Localization

Object centroid localization has been implemented on chip by the authors using parallel hardware [Etienne, 1997, 1998]. The approach taken here has to be compatible with the “computation on read-out” paradigm used in this work. The centroid computation circuit, which is located outside the imaging array, receives either raw image or difference image data. In both cases, a programmable threshold is used to identify the pixels of objects whose centroid is to be computed.

The centroid is computed by loading a bit in a latch at the location where the threshold criterion is met by either the raw image or the difference image. The computation circuit will be realized with two 1D banks of latches representing the x and y axis, respectively. To simplify matters, rows (columns) with many object pixels are not given anymore weight that those with a single pixel. Through this process, a projection of the object is made onto the x- and y-axis. The partitioning of the image array into sub-arrays are performed at the coordinate assignment stage of the computation. When the number of rows corresponding to the size of the sub-array has been scanned, the centroid is determined by finding the average location of latched pixels, similar to the sensor described above. When scanning of the rows of the first group of sub-arrays is complete, the centroid values are output, the latches reset, and the second set of sub-arrays are scanned. Computing the centroid of the entire image is simply done by accumulating all

the pixel that exceed threshold over the entire image, and not dividing the resistive grid and common output line into independent segments. Figure 9 shows a schematic of this advanced centroid localization scheme.

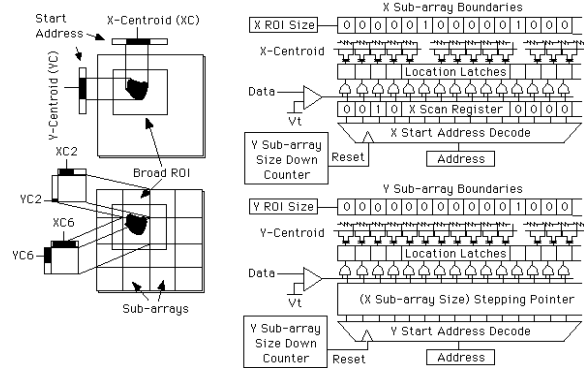


Figure 9: Centroid localization on read-out.

V. DISCUSSIONS AND CONCLUSIONS

The need for and the benefit of having computational sensors in the processing pathway of robot vision systems have been argued. The question on the type of computational sensors to use, i.e. application specific or general-purpose, is somewhat more difficult to answer. To address this issue, an application specific computational sensor for motion centroid localization, modeled after the biological retina and superior colliculus, is presented. Furthermore, two of these sensors, together with a simple μC , are used to realize the solution to a toy line-following with obstacle avoidance autonavigation problem. Despite the simplicity of this problem, this experiment shows that we are currently able to create application specific sensors which can be used to solve real problems in real-time. Generally, application specific sensors are efficient in function, but not in design and re-use. Automation of the design process, using a library of standard cells, can reduce the time-to-product, but this requires a considerable effort to construct this standard library, and software tools to accelerate their silicon compilation. In the area of visual computation and neuromorphic sensors, such a library is beginning to emerge through the networking of designers around the world who make available their designs for use by others. Still there is no organized effort to realize this library.

On the other hand, a general-purpose visual computational sensor would be more useful across

projects and problems. Due to its wide range of functionality, the initial design effort would be very expensive, but not much more expensive than a two or three application specific designs. A general-purpose sensor with programmable functionality would allow the same vision system to be used by many problems and many roboticists. Having a common tool would naturally tighten the research efforts around the world so that the pace of progress would have to increase, as investigators benefit from each other's experience. Furthermore, valuable research time will be reclaimed if the majority manpower initially applied to vision system design is now focused on algorithm development. It is unlikely that a problem will use all the features of the GPCS, however, the cost of the un-used features is acceptable if the benefits of portability are utilized. Hence, the realization of such a GPCS would not be over-kill, even if most of the capability are not utilized in every system. So finally, will this sensor improve the state-of-the-art of robotics? I argue that it must, provided that the *information* provided by the sensor is useful very close to its raw form. If a large amount of post processing is required, then the benefits of information extraction at the focal plane are lost. This implies that the quality of information provided by the sensor must be robotics grade. What is "robotics grade quality?" This specification must be provided by the user. Furthermore, it is application and algorithm specific. So far, the assumption under which we are operating is that 8-10 bits SNR is sufficient. This is realizable with digitally controlled analog processing using VLSI circuits. We are making progress towards a general-purpose computation sensor that will benefit many roboticists attacking a variety of problems.

VI. REFERENCES

- Andreou A., Boahen K., Pouliquen P., Pavasovic A., Jenkins R., and Strohhahn K., "Current-Mode Subthreshold MOS Circuit for Analog Neural Systems," *IEEE Transactions on Neural Networks*, Vol. 2, No. 2, pp. 205-213, 1991.
- Barlow H., *The Senses: Physiology of the Retina*, Cambridge University Press, Cambridge, England, 1982.
- DeWeerth, S. P., "Analog VLSI Circuits for Stimulus Localization and Centroid Computation," *Int'l J. Computer Vision*, Vol. 8, No. 2, pp. 191-202, 1992.
- Etienne-Cummings R., M. Abdel Ghani and V. Gruev, "VLSI Implementation of Centroid Localization for Autonomous Navigation,"

- excepted for publication in *Advances in Neural Information Processing Systems 11*, 1998.
- Etienne-Cummings R., J Van der Spiegel and P. Mueller, "Neuromorphic and Digital Hybrid Systems," *Neuromorphic Systems: Engineering Silicon from Neurobiology*, L. Smith and A. Hamilton (Eds.), World Scientific, 1998.
- Etienne-Cummings, R. and Donghui Cai, "A General Purpose Image Processing Chip: Orientation Detection," *Advances in Neural Information Processing Systems 10*, 1998b.
- Etienne-Cummings R., J. Van der Spiegel and P. Mueller, "VLSI Implementation of Cortical Motion Detection Using an Analog Neural Computer," *Advances in Neural Information Processing Systems 9*, M. Mozer, M. Jordan and T. Petsche (Eds.), MIT Press, pp. 685-691, 1997.
- Etienne-Cummings R., J. Van der Spiegel and P. Mueller, Mao-zhu Zhang, "A Foveated Tracking Chip," *ISSCC97 Digest of Technical Papers*, Vol. 40, pp. 38-39, 1997.
- Horiuchi T., T. Morris, C. Koch and S. P. DeWeerth, "Analog VLSI Circuits for Attention-Based Visual Tracking," *Advances in Neural Information Processing Systems*, Vol. 9, Denver, CO, 1996.
- Horn B., *Robot Vision*, MIT Press, Cambridge MA, 1986.
- Koch C. and H. Li (Eds.), *Vision Chips: Implementing Vision Algorithms with Analog VLSI Circuits*, IEEE Computer Press, 1995.
- Kramer, J. and G. Indiveri, "Neuromorphic Vision Sensors and Preprocessors in Systems Application," *Proc. Adv. Focal Plane Arrays Electronics Cameras*, May, 1998.
- Mansfield, P., "Machine Vision Tackles Star Tracking," *Laser Focus World*, Vol. 30, No. 26, pp. S21-S24, 1996.
- Mead C. (Ed.), *Analog VLSI and Neural Systems*, Addison-Wesley, Reading, MA 1989.
- Mead C. and M. Ismail (Eds.), *Analog VLSI Implementation of Neural Networks*, Kluwer Academic Press, Newell, MA, 1989.
- Mead C. and Mahowald M., "A Silicon Model of Early Visual Processing," *Neural Network*, Vol. 1, pp. 91-97, 1988.
- Mendis S., Kemeny S.E. and Fossum E.R., "CMOS Active Pixel Image Sensor," *IEEE Trans. Electron Devices*, Vol. 41, No. 3, pp. 452, 1995.
- Normann R. and Werblin F., "Control of Retinal Sensitivity," *J. General Physiology*, Vol. 63, pp. 37-61, 1974.
- Sparks D., C. Lee and W. Rohrer, "Population Coding of the Direction, Amplitude and Velocity of Saccadic Eye Movements by Neurons in the Superior Colliculus," *Proc. Cold Spring Harbor Symp. Quantitative Biology*, Vol. LV, 1990.
- Sze S.M., *Physics of Semiconductor Devices (2nd Ed.)*, John Wiley and Sons, New York, 1981.