SPIE Proceedings Vol. 4109, Critical Technoloaies for the Future of Computing,
San Diego, CA, July 30-Aug 4, 2000.

Sensory Computing

Vladimir Brajovic’ and Takeo Kanade

The Robotics Institute, Carnegie Mellon University
Pittsburgh, PA 15213

ABSTRACT

Computation in artificial perceptual systems assumes that appropriate and reliable sensory information about the environment is available. However, today’s sensors cannot guarantee optimal information at all times. For example, when an image from a CCD camera saturates, the entire vision system fails regardless of how “algorithmically” sophisticated it is.

The principal goal of sensory computing is to extract useful information about the environment from “imperfect” sensors. This paper attempts to generalize our experience with smart vision sensors and provide a direction and illustration for exploiting complex spatio-temporal interaction of image formation, signal detectors, and on-chip processing to extract a surprising amount of useful information from on-chip systems.

The examples presented include: VLSI sensory computing systems for adaptive imaging, ultra fast feature tracking with attention, and ultra fast range imaging. Using these examples, we illustrate how sensory computing can extract unique, rich and otherwise not obtainable sensory information when an appropriate balance is maintained between sensing modality, algorithms and available technology.

Keywords: Sensors, Computing, Sensory Computing, Computational Sensors, Smart Sensors, Image Sensors

1. INTRODUCTION

Smart vision sensors integrate image sensing with on chip signal processing. These sensors are interesting because they pose many implementation challenges while offering significant new opportunities. Smart vision sensors are massively parallel systems with tens of thousands of point detectors. They often demand implementation of massive parallelism in a confined pixel space. Local communication with immediate neighbors is readily available, but the naive global information exchange (such as a general bus) across the collection of detectors quickly saturates available wiring. Yet the on-chip processing promises new capabilities and speed.

Making complex pixel processors and then integrating them in a programmable architecture for image processing (such as SIMD – single-instruction-multiple-data’) is an interesting miniaturization of conventional architectures to a chip level, but it results in unreasonably large pixels that find limited applications in high resolution imaging. In addition, many practitioners are finding that the on-chip image processing is not necessary to achieve computational throughput. If reliable image data are available, conventional DSP chips do a reasonable job for demanding machine vision applications.

The wide availability and affordability of the CMOS process has spawned a wave of **CMOS** image sensors. One cited benefit of CMOS vs. CCD imaging is ability of the former to include some processing functions on the same chip[2]. However, the CMOS imaging array itself still suffers from limited dynamic range and will saturate in over- or under-exposed scenes. Artificial perceptual systems (on- or off-chip) that receive such “imperfect” sensory information about the environment will fail in real-world tasks regardless of how “algorithmically” sophisticated the systems are.

The on-chip sensory processing has also spawned the area of neuromorphic chips, sensory chips that mimic biological neural processes’,[4]. Examples include simple implementations of retinal functions such as local contrast encoding and computation of motion flow. This area demonstrated has a myriad of mixed mode circuit solutions whose “unorthodoxy”, creativity and circuit compactness far exceeds that of all other smart sensor solutions[5].

Obviously, integrating image sensors and processors in a single chip is not a new idea. While all of these ideas employ computation at a sensory level, they do not necessarily perform sensory computing. ***The** principal goal of sensory computing,*

[1] Correspondence: tel.: 412-268-5622, email: brajovic@cs.cmu.edu, **http://www.cs.cmu.edu/–brajovic**

as we define it, is to extract useful, rich arid otherwise not obtainable environmental information from "imperfect" detectors. Although miniaturization and increased computing power usually follow as natural benefits of sensory computing, its primary goal is to make better use of available detectors.

This paper summarizes our experience in computational sensors and attempts to outline a "guideline" for building superior sensors using sensory computing. We will illustrate our points with our own examples and will mention examples of others. This illustration is by no means exhaustive and many unmentioned cases will find that their implementation does involve a degree of sensory computing.

We briefly describe the features of on-chip computation that benefits sensory computing. Then we define dimensions to be explored when finding creative space-efficient circuits and architectures. Finally, we illustrate how these guidelines are useful in several implementations of sensory computing.

1.1. Benefits of Sensory Computing

Top-down adaptation at the sensory level extends the capability of native detectors. Pushing for a brute force sensitivity of detectors is one level of sensory improvement. But however sophisticated these detectors may be, the environmental conditions always demand more, thereby rendering these detectors "imperfect". The sensory computing makes decision using sensory signals and can adapt detectors properties before the environmental stimulus exceeds the detectors' native capabilities.

Parallelism over collection of point detectors extends the capabilities of any single detector. Groups of single-point detectors, such as pixels in image sensors, detect broader environmental context and provide broader support for more robust adaptation. The sensory computing may influence each detector's value by signals from other detectors. In the extreme, all point detectors "agree" on what is the "best" image to report, thus creating a whole whose functionality is greater than the sum of its parts.

Sensory computing enables low-latency parallelism. Low latency, or quick reaction time, is necessary for closing the adaptation loop quickly. On-chip parallelism, if implemented correctly, will deliver speed, thus providing information for low latency decisions in time-critical applications. The latency is the most difficult characteristic to scale in parallel systems. DSP chips can deliver desired throughput measured by the data update rates at input and output. The low latency, however, can be best achieved with sensory computing.

From our experience, the only time we will want to invest time and effort in designing computational sensors is when the sensory signal is not reliable, or when we need low system latency. These two requirements often go hand in hand: to achieve sensory adaptation we need low system latency.

1.2. Dimensions of Sensory Computing Implementation

To deliver superior, yet practical, performance in a small pixel and high spatial resolution computational sensors must be designed not only to take advantage of available signal processing means, but also, depending on the application, to leverage and manipulate the formation of sensory stimuli and transduction of signals.

Stimulus transduction spans dimension of time. An optical stimulus is a flux of photons. It is modeled as a Poisson arrival process whose rate of arrival is controlled with general "intensity" of the stimulus. A photo detector can operate in instantaneous mode in which it reports the level of the photon flux, including the shot noise of the stochastic arrival process. When a photo detector operates in a flux-integration mode[6], it integrates photoelectrons, thus reducing (averaging) the shot noise. In the conventional image sensors, the integration process continues for a fixed integration period, followed by the measurement of accumulated charge in each pixel. If the photon flux at some pixels exceeds the limits of the sensor, the photo-charge saturates those pixels. Shortening the integration period is the usual remedy for this problem, but at the expense of not collecting sufficient photo-charge at the dark pixels. This is a well-known limited dynamic range problem of conventional image sensors; a problem that right at the acquisition level limits the entire vision system. In addition to simply dealing with the amount of charge collected during **a** fixed integration interval, the sensory computing can deal with the time intervals the detector takes to accumulate a predetermined amount of photo-charge.

Global and local parallel computation in dimension of space. So far, a great majority of computational sensory solutions implement local operations on a single light sensitive VLSI chip. Local operations use operands within a small spatial/temporal neighborhood of data and thus lend themselves to graceful implementation in VLSI. Typical examples include filtering and motion computation. Local operations produce preprocessed "images;" therefore, a large quantity of data still must be read out and further inspected before a decision for an appropriate action is made — usually a time-consuming process that creates large latency. Locally computed quantities could be used for adaptation within the local neighborhood, but for global adaptation the latency is excessive. Consequently, a great majority of computational sensors built thus far are limited in their ability to quickly respond to changes in the environment and to globally adapt to new situations. Global operations, on the other hand, produce fewer quantities for the description of the environment. An image histogram is an example **of** a global image descriptor. If computed at the point of sensing, global quantities can be routed off a computational sensor through a few output pins without causing a transfer bottleneck. In many applications, this information will often be sufficient for rapid decision-making and the actual image does not need to be read out. The computed global quantities also can be used in top-down fashion to update local *and* global properties of the system for adapting to new conditions in the environment. Implementing global operations in hardware, however, is not trivial. The main difficulty comes from the necessity to bring together, or aggregate, all or most of the data in the input data set. This global exchange of data among a large number of processors quickly saturates communication connections and adversely affects computing efficiency in parallel systems — parallel digital computers and computational sensors alike.

Sensory computing includes the dimension of stimulus formation. An optical system forms an image on the sensor plane. The optics determines the optical preprocessing of the scene. For example, defocusing the optics performs crude low-pass filtering. Sometimes we are able to influence the environment by engineered illumination. Illuminating scenes with structured light, has been the standard technique for providing salient image feature that would aid in subsequent processing. The more conveniently the stimulus is formed, the simpler and more capable the on-chip processing will be. That is important because we are limited in space and capabilities of on-chip processing.

2. EXPLOITING DIMENSION OF TIME IN PHOTOSENSING

We have implemented two analog VLSI computational sensors for sensing and encoding high dynamic range images by exploiting temporal dimension of photoreception. In addition to simply dealing with the amount of charge collected during a fixed integration interval, we deal with time interval required by the detector to accumulate a predetermined amount of photo-charge. Both sensors have been reported previously[7,8,9]. Here we briefly outline their function for illustrating the benefits of exploiting the dimension of time in computational sensing.

2.1. Multiple Integration Time Photoreceptor

The first sensor is a multiple integration time photoreceptor that avoids saturation by automatically choosing one integration interval from a set of predetermined intervals. When the charge level becomes close to saturation, the integration is stopped at one of these integration periods. The receptor encodes the intensity with two signals: 1) the accumulated charge, and 2) the identifier of the selected interval. The sensor can represent a wide range of light intensities using these two signals.

Figure 1 shows the circuit for the MIT photoreceptor, and Figure 12 shows the representative signal waveforms. The receptor includes two photodiodes operated in the photon flux integration mode. An inverter thresholds the voltage of the photodiode A (Pd-a) and is responsible for detecting saturation. A transparent latch controlled by the train of timing pulses (New-IT) acquires the output of the inverter during each pulse. The periods between consecutive pulses define the set of available integration periods. Each subsequent period is twice as long as the previous one. In **our** experiments, we used eight integration times per frame; the Nth integration period is a 1/(28-N) of one frame time.

The photodiode B (Pd-b) integrates signal charge. The output of the latch samples and holds this charge on storage capacitor Capl. The latch output thus controls the duration of photon integration.

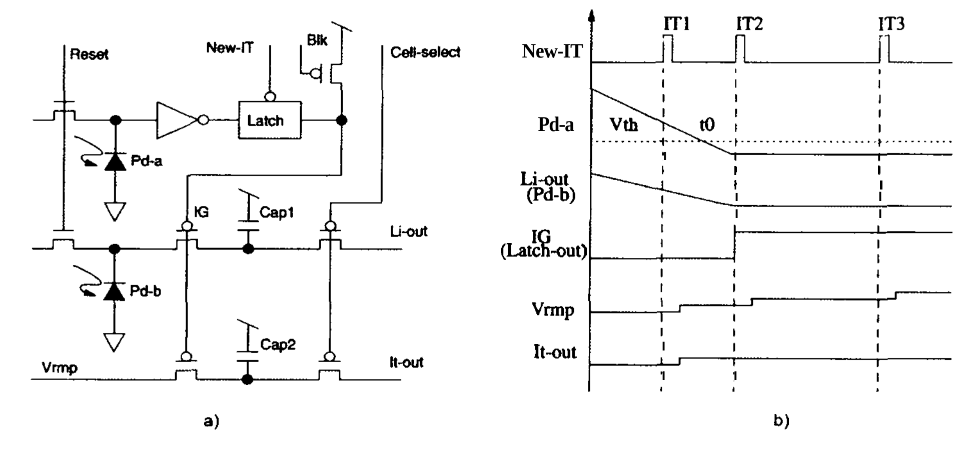

Figure 1 : Multiple integration time photoreceptor: a) pixel circuit, b) signal waveforms

The latch output also samples and holds a ramp voltage in the capacitor Cap2 to memorize the identity of the period at which the Pd-b stopped integrating. The ramp voltage is incremented by a predetermined step at each pulse in New-IT; therefore, it indicates the number of the interval being used.

Figure 2 shows an early part of a frame. In this example, the illumination is such that the second integration period is chosen. Following the reset to a high voltage, the two photodiodes integrate the signal charge. Pd-a decays, passes the first integration timing (It1), and then reaches the threshold for the inverter at to. The inverter trips from low to high. However, the integration gate (IG) dose not change because the new state of the inverter output is still not visible at the latch output. Only after the integration timing pulse It2 makes the latch transparent for a short time (e.g., Ims) the new state of the inverter affects the sample/hold gate IG. Then, the integration in Capl stops and the identity of the It2 is recorded. The blanking gate Blk forces the integration gate (IG) to open at the end of the frame if the light intensity is too low to do so earlier.

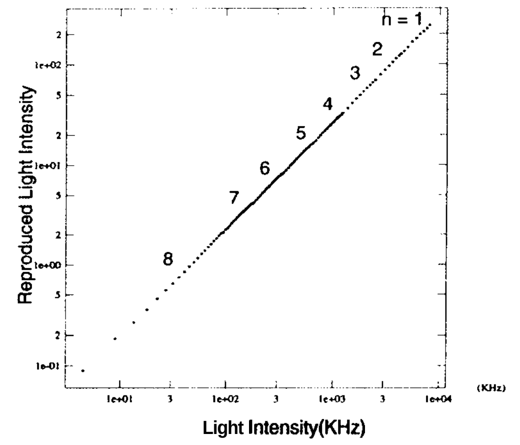

Figure 2: Reconstructed wide dynamic range signal for multiple integration time photoreceptor. The varying input illumination is generated by pulsating **LED.**

From the two signals, Li-out and It-out, we can reconstruct the actual light intensity by multiplying Li-out by 2(8-N), where N is the value of It-out. The reproduced outputs are plotted against light intensity in Figure 5. There are eight segments of this graph, corresponding to the different integration periods. The transition from one integration period to the next is smooth and the output as a whole shows very good linearity over a wide range. The smallest integration period is one 128th of the largest integration time; therefore, the dynamic range is 128 times larger than that of a single integration period photoreceptor, for approximately 1:128,000 dynamic range.

2.2. The Sorting Image Sensor

The second method, a sorting image sensor, avoids saturation by sorting pixels according to their intensities. The input image may have a large (or low) dynamic range, but the indices assigned to pixels always range from 1 to N, where N is number of pixels in the imager. The sorting in the sensor is based on the biologically inspired notion that stronger stimuli elicit responses before weaker ones". The sorting is achieved in analog by dealing solely with the time intervals required by each receptor to accumulate a predetermined amount of photo-charge. A block diagram of our implementation is shown in Figure **3.** A detailed schematic can be found elsewhere'. In **our** implementation, each pixel integrates a charge until a predetermined amount is accumulated. Then, an event is fired. The temporal integration ensures that the brighter pixels fire their events before the darker ones. Therefore, the pixel events are ordered in time according to their intensities. An analog global counter tallies the events. When the first response is received, the global count is one. This count represents the order, or index, of the cell that generated the event. The sorting of input signals is thus achieved by assigning the global count to the cell that generated the most recent event. The global count is fed back to all pixels. The pixel that fired the most recent event memorizes this count as its index. For example, when the second cell responds, the global count is two, which is then assigned as an index for the second cell, and so on. The more time allowed, the more responses are received; thus, the global counter incrementally accounts for all pixels in the array. At the end, each pixel contains its own index — an image of indices. This method of sorting is closely related to a counting sort for integers".

Readers may recognize that the image of indices is a histogram-equalized version of the original image. The evolving cumulative count is the temporal representation of the cumulative histogram of the detected **image**[12]. The cumulative histogram **is** one global property of the scene that is reported by the chip with very low latency, and can be used for preliminary decision-making as soon **as** the first responses are received.

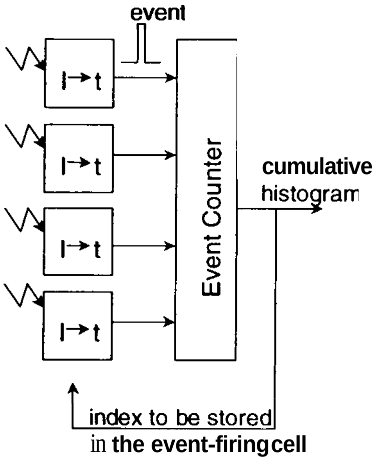

Figure *3:* Block diagram of the sorting image sensor. Only four pixels are shown.

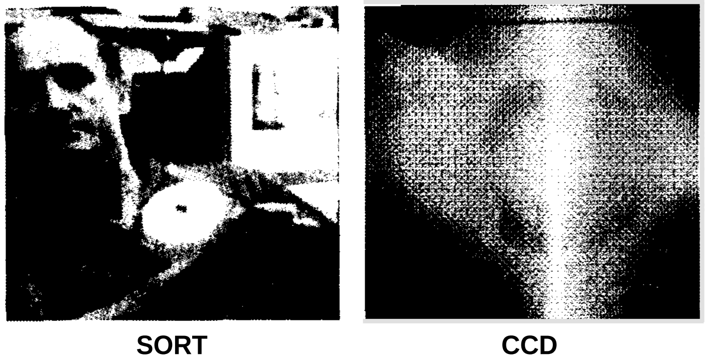

Figure 4: Adapting to high dynamic range scenes: Sorting image sensor (left), CCD image sensor (right). The linear image of the scene is reconstructed from the image of indices and the cumulative histogram waveform. The reconstructed image is scaled from about 18-bits per pixel to 8-bits per pixel to accommodate document printing.

The sensor also uses the cumulative histogram waveform to map detected light intensities into indices. This waveform can be used for mapping indices back to the received intensities. Therefore. the sorting sensor encodes large dynamic range images with I) the image of indices, and 2) the cumulative histogram waveform. The image of indices has uniform histogram indicating that the indices are equiprobable. Therefore, when storing and transmitting the image of indices, the sensor uses the available signal–to–noise ratio most efficiently; the image of indices is information-theoretic optimal representation[13]. Figure 4 shows imaging of a high dynamic range scene, while Figure 5 shows imaging of a low-contrast scene

2.3. Local vs. Global Data Aggregation

The multi integration time photoreceptor works well as a single point detector. Recently, a full scale imaging array using this concept has been reported". This approach would deliver a plurality of independent pixel measurements each independently adapting to

Our two methods efficiently encode sensory information. They are practical because the encoding is easy to interpret and the noise introduced by processing **is** minimal. Both sensors have pixels whose size **is** less than 30um x 30um.

3. EXPLOITING DIMENSION OF SPACE

We built a computational sensor for optical tracking that focuses attention on a local intensity peak in its field of view by using self-adapting spatial selectivity. Using both low–latency massive parallel processing and top–down sensory adaptation, the sensor suppresses interference from features irrelevant for the task at hand, and tracks a target of interest at speeds of up to 7000 pixels/second. The sensor locks onto the target to continuously provide control **for** the execution of a perceptually guided activity. The sensor prototype is a 24x24 array of cells. Each cell occupies 62um x 62um of silicon, and contains a photo detector and

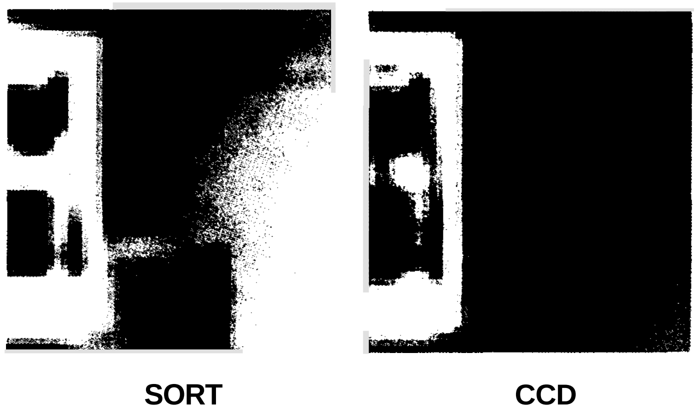

Figure 5: Adapting to low contrast scenes. The low-contrast shading on the background wall is greatly enhanced in image of indices.

processing electronics. The details of the sensors have been previously reported[21].

Our tracking computational sensor optically receives an image, selects a peak in that input image, and continuously reports the location and magnitude of the selected peak. In the context of this paper, the selected point is called a target. The location of the target is global information that is reported as the output. The location of the target is also used internally to self-adapt the location of the attention to implement target locking.

An image is optically focused onto the array of photo detectors. Generated photocurrents are fed to the winner–take–all (WTA) circuit[22.23], which is responsible for the feature selection. The WTA circuit also reports the intensity of the winner on one globally accessible wire. The cells of WTA are organized in a two–dimensional array.

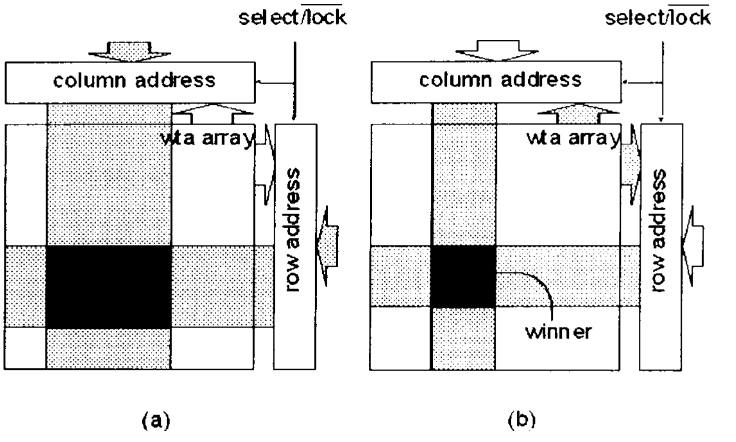

Figure 6: Select (a) and lock (b) mode of the tracking computational sensor.

The WTA circuit locates the absolute maximum in the entire image. In practical applications, there are often several targets in the scene. The target of interest is not necessarily the strongest. We need to direct the sensor's attention toward that target. Once the target is selected, we need a mechanism that will lock and track the target while the target is of interest and/or a perceptually guided goal is being executed. **Our** implementation solves these issues by inhibiting a portion of the saliency map, thus restricting the activity of the WTA circuit to a programmable active region — a subset of the array. Appropriate row-column addressing programs the active region (see Figure 6). There are two modes of operation: (1) select mode, and (2) lock mode. In the select mode, the active region is user–defined by the external addressing (Figure 6a). The active region is of arbitrary size and location. The target selected by the sensor is the absolute maximum within this region. In lock mode, the sensor itself dynamically defines a small (e.g., 3 x 3 in this implementation) active region centered at the most recent location of the target (Figure 6b). The select mode directs the attention toward a feature that is useful for the task at hand. For example, a user may want to specify an initial active region, aiding the sensor in attending to the relevant local peak in the scene. Then, the lock mode is enabled for locking onto the selected feature. In the lock mode, the 3x3 cell active region is centered at the location of the current attention target. If the target moves, one of the eight active neighbors in the WTA array will receive the winning intensity peak and automatically update the position of the 3x3 active region. It is now clear that the salient target is not necessarily the peak of the absolute maximum intensity in the image. The ability of the sensor to define its own active region is an example of the top-down sensory adaptation presently missing in conventional machine vision systems.

The robust performance of the sensory attention and sensor's select/lock feature is illustrated in Figure 7, which shows a CCD camera image of the scene of this experiment. There is a horizontally scanned laser dot and an arbitrarily roaming dot produced by a hand–held laser pointer. (The scanning dot appears as a line, since the scanning frequency exceeds the speed of the conventional CCD camera). Figure 7 also shows the oscilloscope displays of the tracking sensor signals. When there is no spatial sensory selection, the WTA array exhibits unreliable behavior; the two targets interfere with each other, and the sensor erratically

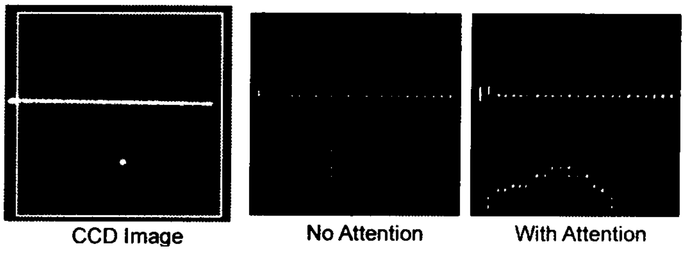

Figure 7: Tracking performance

The proposed VLSI implementation of the tracking computational sensor exhibits several interesting features. It senses input images and produces a few global results: the position and magnitude of the target being tracked. With no latency, these global results are reported off–chip via few output pins. Furthermore, in the lock mode, the global results are used internally for programming a 3x3 active region, thus providing a low–latency top-down spatial adaptation for securing robust performance in a rapidly changing environment. Such an adaptation, and hence reliable performance, is currently missing in conventional machine vision systems. In our implementation, the sensor robustly tracks targets moving up to 7000pixels/second, while consuming only 0.25mW of static power. If we assume that a conventional CCD operates at 30 frames per second, and that its pixels are of about 10um size, we can convert the speed of 7000 pixels/second to about 2000 CCD pixels per CCD frame. Since the conventional CCD arrays measure about 750 pixels across, this measure means that the target we are tracking would scan about 3 times across the CCD's field of view within one frame. Clearly in this case. the low-latency on-chip processing demonstrates performance that is not replicable by conventional systems.

4. EXPLOITING THE DIMENSION OF IMAGE FORMATION

The issues addressed by the tracking sensor are analogous to issues facing the implementation of rudimentary visual attention[24]. The tracking computational sensor implements primitive attention. Bright spots in received images are considered salient and are potential targets for tracking. If a particular saliency such as color or a particular intensity pattern is of interest then optical (or electronic) preprocessing is **needed**[25,26,27]. In general, the input images to the tracking chip can be considered to be optical saliency maps that encode "conspicuousness" of targets through the scene. Broad-spectrum intensity images used in our experimentation are trivial saliency maps.

Now we move on to our recently developed computational sensor for rapid range imaging. It exploits complex spatio-temporal interaction between image formation setup and on-chip processing. Measuring range and the three–dimensional (3D) profile of objects is important in many applications. Triangulation–based light stripe methods are the most practical and quite robust[28]. The triangulation setup is shown in Figure **8.** The stripe of laser light is projected onto the scene. **A** sensor, usually a CCD camera, views the scene. The depth to the point on the object is found as:

$$z_o = \frac{B}{\frac{x_o}{f} + \tan\alpha} \tag{1}$$

where ***B*** is the baseline separation of the laser and the camera optical centers, *f* is the focal length of the camera lens, x_0 is the image location within a row where the laser stripe is detected, and α is the projection angle of the laser in respect to the *z* axis. In order to ease the detection of the laser in the image, the illumination conditions are usually adjusted so that the projected laser generates the brightest features in the scene. Traditionally, the range map is collected one slice at a time. The laser stripe is fixed at a particular angle ***a,*** the scene is imaged with a CCD camera, and the x_0 location is detected in each row. Then, the laser stripe is repositioned and another slice of the range map is collected. This process is too slow: each slice requires at least one camera frame time.

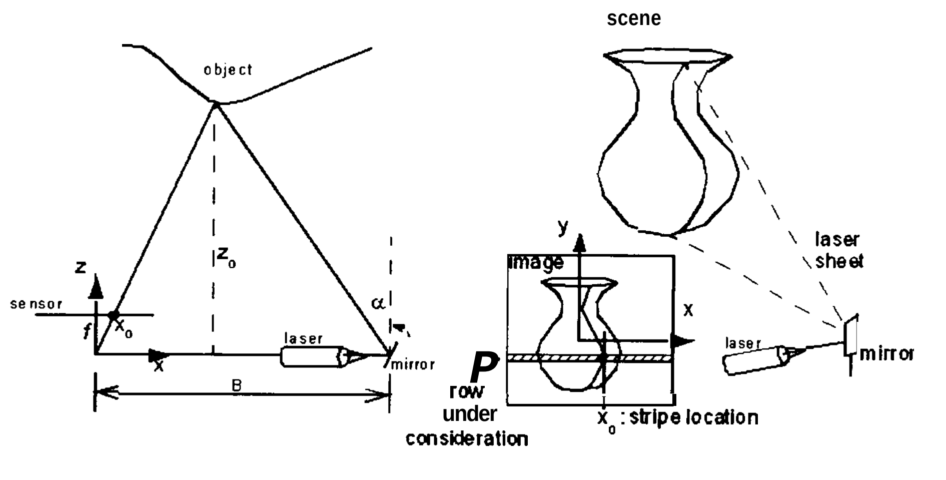

Figure **8:** Image formation setup for dynamic triangulation. The sheet of laser light **is** swept across the scene.

A high-speed triangulation approach has been **proposed**[29]. In this method, the laser in Figure I is continuously swept across the scene, say from right to left. Each pixel in the sensor has its own line of sight and "sees" the laser stripe only once as it sweeps by. By recording the time when a particular pixel at location xo sees the laser, the range is calculated as:

$$z = \frac{B}{\frac{x_o}{f} + \tan \omega t} \qquad (2)$$

where ω is the angular velocity of the mirror. This technique has been implemented in two cell–parallel VLSI range sensors[30,31]. In the first sensor[30] each cell detects the temporal intensity peak; at that time, the cell records its time–stamp. In the second sensor[31] each cell includes two photo–detectors. The stripe is detected when the appreciable difference between the two photocurrents is observed; then a time–stamp is associated with each cell.

Even though these cell–parallel VLSI implementations work well, the triangulation method is inherently row–parallel. Only one pixel in each given row sees the laser stripe at any given time. Therefore, the detection of the image of the laser stripe is a global operation over each row of pixels. We developed a row–parallel architecture employing a winner–take–all (WTA) circuit embedded in each row of the sensor. The WTA in each row detects the location of the pixel receiving the most light, i.e., the image of the laser stripe. Therefore, most of the circuitry in [30,31] can be removed from each cell and reused once per row on the side of the array. The proposed architecture builds on our experience with the tracking sensors[2'].

The sensor is a 2D array of photodetectors with an embedded WTA circuit in each row. The sensor optically receives an image, selects an absolute intensity peak in each row, and reports the location and magnitude of the selected peak. The location of the peak and its intensity is the global row information that is reported as the output for each row.

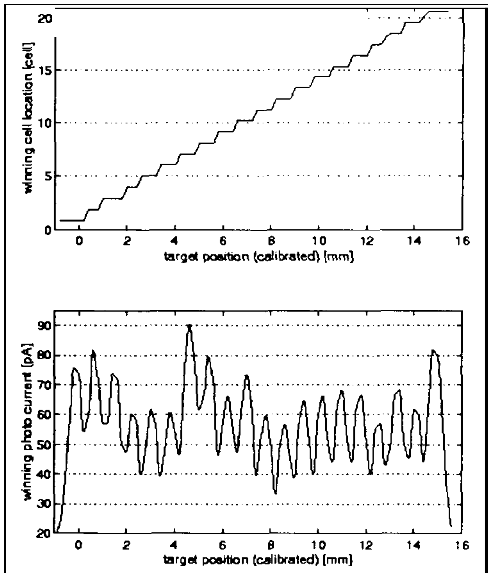

Figure 9: When the target moves across the row of sensors, the **WTA** circuit reports pixel location and the intensity of the winner. The peak in the intensity waveform occurs when the target is centered on a photodetector. The valleys occur when the target transitions **from** one photodetector to the next.

Figure 9 shows these two signals as the bright laser line travels across the row of 20 photodetector. A particular cell remains a winner as long as the main portions of the bright target are focused on it. This appears as the staircase line (Figure 9, top graph). As the stripe is moved, its image leaves one cell and begins contributing photocurrent to the next one. At some point, the cell receiving the target wins and takes control of the common wire reporting the winner's intensity (Figure 9, bottom graph). As the stripe moves toward the center of the new winning cell, the intensity of the winning input current increases. The cell continues to win as the stripe passes the center, but its input current begins to diminish. In the meantime, the next cell begins to receive an increasing amount of light and the process continues. Therefore, as the target passes over the winning cell, the measured common voltage increases, peaks, and then decreases. This behavior is clearly displayed in the bottom graph **of** Figure 9.

The positive peak occurs when the target is centered on the photodetector. Conversely, the negative peak occurs when the target is positioned exactly between two photodetectors. Therefore, locating peaks (positive and negative) allows precise localization **of** the target at the spatial grid that is twice the resolution of the photodetectors.

The row–parallel architecture of the sensor is shown in Figure **10.** The WTA circuit continuously localizes the laser stripe in the row and generates on its common wire a temporal voltage waveform similar to one shown in the bottom graph of Figure 9. A one per row peak detector monitors this voltage. When the peak detector detects either a positive or negative peak in a particular row, it latches the time in one-per-row memory. The row memory locations are rapidly scanned. All rows are scanned several times within the time it takes the laser to travel across one photodetector. In this way, we ensure that no new peaks occur before we read out information regarding the previous peaks. When the scanner selects a row, the WTA is multiplexed to the position encoder. At the same time, the type of the peak (i.e., positive or negative) is also multiplexed to the output. Note that the address and the *peak* type uniquely determine the exact position **of** the stripe. If the peak is positive, the stripe position is in the center of the photodetector whose address is being reported. If the peak is negative, the stripe position is on the (stripe) receiving edge of the photodetector whose address is being reported. Even though the cell address is not well defined when the stripe transitions to a new cell, the propagation delay through the multiplexer allows the stripe to move into the receiving cell, thus ensuring a stable address. During the readout. the scanner reports the address of the selected row.

At the beginning of each frame, a timer is reset. Time information, together with the address and the peak type, provides all the information needed for the reconstruction of the range map according to Equation *(2).*

The triangulation range mapping has several advantages over previous VLSI implementations:

- Most of the computational circuitry is moved out of the cell and placed at the edge of the array. This results in a high spatial resolution; the prototype has

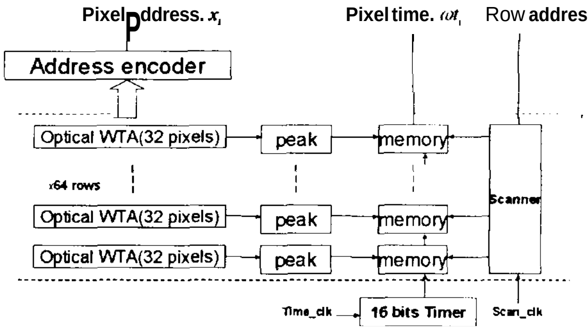

Figure 10: Range sensor architecture

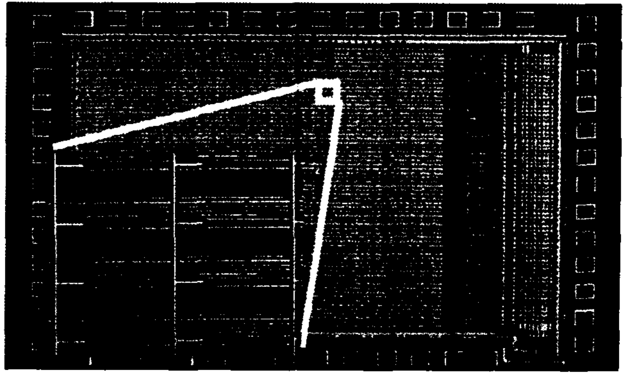

Figure 12: Layout of the range sensor. The inset shows 6 pixels with 2-to-1 aspect ratio each.

In conclusion, the range sensor demonstrates how a surprisingly powerful performance with low system complexity can be achieved when the image formation and the on-chip

5. CONCLUSION

This paper stresses that the main goal of sensory computing is not to miniaturize conventional system, but rather to extract useful information from available "imperfect" detectors. Interesting implementations of sensory computing will have superior sensing capabilities compared to conventional sensors and will have no substitutes in conventional systems. From our experience, the ability to adapt detections based on low-latency results provided by the on-chip processing is the main feature that warrants the implementation of sensory computing. With

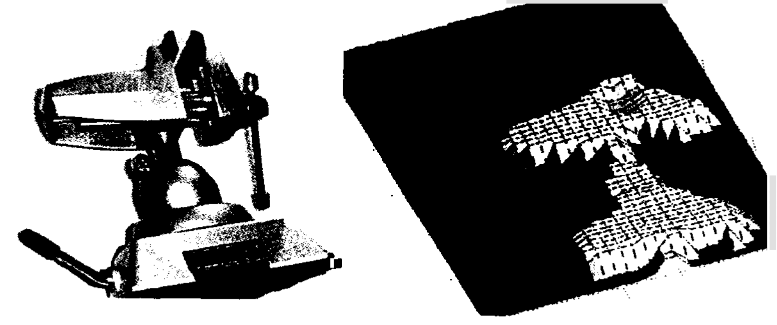

Figure 11: A range map of an object collected by the range sensor.

present integration technology, new processing paradigms that are able to exploit complex spatio-temporal interaction of stimulus formation, stimulus detection and on-chip processing in a confined space of a small pixel are needed. The ultimate test for successful implementations ought to be whether they are practical in real-world applications.

ACKNOWLEDGMENT

This research has been partially funded by the ONR Grant N00014–95–1–0591 and by the NSF, Grant MIP–9305494. We are grateful to Mr. Kenichi Mori for his work on the range sensor, Dr. Nebojsa Jankovic for testing the latest generation sorting sensor, and Mr. Ryohei Miyagawa for designing and testing the multiple integration time photoreceptor.

REFERENCES

1. Masatoshi Ishikawa, Kazuya Ogawa, Takashi Komuro, and Idaku Ishii: A CMOS Vision Chip with SIMD Processing Element Array for 1ms Image Processing, 1999 Dig. Tech. Papers of 1999 IEEE Int. Solid-state Circuits Conf. (ISSCC'99) (San Francisco, 1999.2.16) pp.206-207
2. **S.** K. Mendis, **S.** E. Kemeny, **R.** C. Gee, B. Pain, C. O. Staller, Q. Kim and E. R. Fossum, "CMOS active pixel image sensors for highly integrated imaging systems", ***IEEE J. Solid-State Circuits,*** vol.32, pp. 187-197, Feb. 1997.
3. Mead, C. (1989). Analog VLSI and Neural Systems. Reading, MA: Addison-Wesley.
4. Mathur, B. & Koch, C. (Eds.) (1991). Visual Information Processing: From Neurons to Chips. Proc. SPIE, 1473
5. Alireza Moini. Vision Chips, Kluwer Academic Publishers, 1999.
6. Weckler. G.P. (1967). Operation of p-n Junction Photodetectors in a Photon Flux Integrating Mode. IEEE Jour. of Solid–State Circuits, 2, 65–73, September 1967.
7. Brajovic, V. and Kanade, T. (1996). A Sorting Image Sensor: An Example of Massively Parallel Intensity-to-Time Processing for Low-Latency Computational Sensors. Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, MN, 1638–1***643.***
8. V. Brajovic and T. Kanade, "A Sorting Image Sensor: An Example of Massively Parallel Intensity-to-Time Processing for Low-Latency Computational Sensors," Proceedings of the 1996 IEEE International Conference on Robotics and Automation, April 1996, Minneapolis, MN.
9. V. Brajovic, R. Miyagawa and T. Kanade, "Temporal Photoreception for Adaptive Dynamic Range Image Sensing and Encoding," Neural Networks **11,** pp. 1149-1158, 1998
10. Ripps, H. & Weale, R.A. (1976). Temporal Analysis and Resolution. In Davson, H. (Ed.) The Eye. 2A: Visual function in man. New York: Academic Press, pp. 185-217.
11. Cormen, T.H., Leiserson, C.E. & Rivest R.L.(1992). Introduction to Algorithms. Cambridge, MA: MIT Press.
12. Ballard, D.H. and Brown, C.M. (1982). ***Computer Vision.*** Englewood Cliffs, N.J.: Prentice-Hall, 1982.
13. Shannon, C. & Weaver, W. (1949). The Mathematical Theory of Communication, Urbana, IL: University of Illinois Press.
14. S. Benthien, T. Lule, B. Schneider, M. Wagner, M. Verhoeven and M. Bohm, "Vertically Integrated Sensors for advanced Imaging Applications," ***IEEE J. Solid-state Circuits,*** vol.35, pp. 939-945, July 200.
15. Brajovic, V. (1996). Computational Sensors for Global Operations in Vision. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania.
16. M. Schanz, C. Nitta, A. Busmann, B.J. Hosticka, R.K. Wertheimer, "A High-Dynamic-Range CMOS Image Sensor for Automotive Applications," ***IEEE J. Solid-state Circuits,*** vol.35, pp. 932-938, July 200.
17. S. Decker, R. Mc Grath, K. Brehmer and C. Sodini, "A 256 x 256 CMOS imaging array with wide dynamic range pixels and column-parallel digital output", in ***Dig. Tech. Papers IEEE Int. Solid-state Circuits Conf.*** , Feb. 1998, pp. 176-177.

18. Boahen, K.A. & A.G. Andreou (1992). A contrast sensitive retina with reciprocal synapses. In J.E. Moody, S.J. Hanson & R.J.P. Lippmann. (Eds.), Advances in Neural Information Processing Systems 4 (pp. 764-772). San Mateo, CA: Morgan Kaufmann Publishers.

19. Delbruck, T. and Mead, C.A. (1994). Analog VLSI Phototransduction. California Institute of Technology, CNS Memo No **30.** May 11, 1994.

20. Yamada, K. Tomoaki N., Yamamoto. S. (1997). Effectiveness of Video Camera Dynamic Range Expansion for Lane Detection. Proc. of IEEE Conference on Intelligent Transportation Systems, Boston, MA, November 1997.

21. V. Brajovic and T. Kanade, "Tracking Computational Sensors with Attention," to appear in IEEE JSSC, August 1998.

22. A.G. Andreou, et al., "Current-Mode Subthreshold MOS Circuits for Analog VLSI Neural Systems," IEEE Transactions on Neural Networks, Vol. 2, No. **2,** March 1992, pp. 205-213.

23. J. Lazzaro, S. Ryckebusch, M.A. Mahowald and C. Mead, "Winner-Take-AI1 Networks of O(n) Complexity," in Advances in Neural Information Processing Systems Vol. 1, D. Tourestzky, ed., pp. 703-711, Morgan Kaufmann, San Mateo, CA, 1988.

24. A. Allport, "Visual Attention," Foundation of Cognitive Science, M. Posner, ed., MIT Press, 1989, pp. 631–682.

25. C. Koch and S. Ullman, "Shifts in Selective Visual Attention: Toward the Underlying Neural Circuitry. In L.M. Vaina, ed., Matters of Intelligence, Reidel Publishing, 1987, pp. 115-141.

26. **T.K.** Horiuchi, T.G. Morris, C Koch. and S.P. DeWeerth, "Analog VLSI Circuits for Attention–based, Visual Tracking," Advances in Neural Information Processing Systems, Vol. 9. MIT Press, 1997.

27. T.G. Morris and S.P. DeWeerth, Analog VLSI Circuits for Covert Attentional Shifts, MicroNeuro '96, Lausanne, Switzerland.

28. P.J. Besl, "Range Imaging Sensors," Res. Publ. GMR-6090, GM Research Labs., Warren, MI, Mar. 1988.

29. Y. Sato, K. Araki, and S. Parthasarathy, "High speed rangefinder," in Optics, Illumination, and Image Sensing for Machine Vision II, SPIE, vol. 850, pp. 184-188, 1987.

30. **A.** Gruss, L.R. Carley and T. Kanade, "Integrated Sensor and Range-Finding Analog Signal Processor," IEEE JSSC, vol. 26, No. **3,** March 1991.

31. A. Yokoyama, K. Sato, T. Yoshigahara, **S.** Inokuchi, "Realtime Range Imaging using Adjustment-Free Photo-VLSI – Silicon Range Finder," Proceedings IROS, pp. 1751-1758, 1994