IEEE Transactions on Robotics and Automation,
submitted, July 1997

Accepted as regular paper: G97179

A VLSI Sorting Image Sensor: Global Massively Parallel Intensity–to–Time Processing for Low–Latency, Adaptive Vision

Vladimir Brajovic and Takeo Kanade

The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

voice: 412-268-5622
fax: 412-268-5571
email: brajovic@cs.cmu.edu

Abstract. This paper presents a new intensity–to–time processing paradigm suitable for VLSI computational sensor implementation of global operations over sensed images. Global image quantities usually describe images with fewer data. When computed at the point of sensing, global quantities result in a low–latency performance due to the reduced data transfer requirements between an image sensor and a processor. The global quantities also help global top–down adaptation: the quantities are continuously computed on–chip, and are readily available to sensing for adaptation. As an example, we have developed a sorting image computational sensor — a VLSI chip which senses an image and sorts all pixels by their intensities. The first sorting sensor prototype is a 21 by 26 array of cells. It receives an image optically, senses it, and computes the image's cumulative histogram — a global quantity which can be quickly routed off chip via one pin. In addition, the global cumulative histogram is used internally on–chip in a top–down fashion to adapt the values in individual pixels so as to reflect the index of the incoming light, thus computing an "image of indices." The image of indices never saturates and has a uniform histogram.

1. Introduction

Many time–critical robotics applications, such as autonomous vehicles and human–machine interfaces, need a *low–latency* and *adaptable* vision system. Conventional vision systems comprised of a camera and processor provide neither low–latency performance nor sufficient adaptation.

Latency is the time that a system takes to react to an event. In the conventional systems, the latency is incurred in both the data transfer bottleneck created by the separation of the camera and the processor and in the computational load bottleneck created by the necessity to process a huge amount of image data. For example, a standard video camera takes 1/30 of a second to transfer an image. In many critical applications, the image capture alone presents excessive latency for the stable control of a robotic system. Another example is the pipelined dedicated vision hardware which delivers the processing power to update its output 30 times per second; however, the latency through the pipeline is typically several frame times, again rendering the conventional system unsuitable for many time–critical applications.

The second important feature of a vision system is adaptation. It has been repeatedly observed in machine vision research that using the most appropriate sensing modality or setup, allows algorithms to be far more simple and reliable. For example, the concept of active vision proposes to control the geometric parameters of the camera (e.g., pan, tilt, zoom, etc.) to improve the reliability of the perception. It has been shown that initially ill–posed problems can be solved after the top–down adaptation of the camera's pose has acquired new, more appropriate image data [1]. Adjusting geometric parameters is only one level where adaptation can take place. A system which can adjust its operations at all levels, even down to the point of sensing, would be far more adaptive than the one that tries to cope with the variations at the "algorithmic" or "motoric" level alone.

The computational sensor paradigm [11] [3] has potential to greatly reduce latency and provide adaptation. By integrating sensing and processing on a VLSI chip, both transfer and computational bottlenecks can be alleviated: on–chip routing provides high throughput transfer, while an on-chip processor could implement massively parallel computational models. Adaptation is also more conveniently facilitated: the results of processing are readily available to sensing for adaptation.

So far, a great majority of computational sensory solutions implement *local operations* on a single light sensitive VLSI chip (for examples, see [11] [13] [19]). 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. Locally computed quantities could be used for adaptation within the local neighborhood, but not globally. 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 a new situation.

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 [5] [3]. 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. It is not surprising that there are only a few computational sensors for global operations: [18] and [8] show computational sensors for detecting isolated bright objects against the dark background and computing the object's position and orientation, and [16] and [6] show attention–based computational sensors that globally select the "most salient features" across the array of photoreceptors.

This work introduces a novel intensity–to–time processing paradigm — an efficient solution for VLSI implementation of low–latency massively parallel *global* computation over large groups of fine–grained data. By using this paradigm, we have developed a sorting computational sensor — an analog VLSI chip which sorts pixels of a sensed image by their intensities. The sorting sensor produces *images of indices* that never saturate. As a by-product, the chip provides a cumulative histogram — a global descriptor of the scene— on one of the pins; this histogram can be used for low–latency decision making *before* the image is read out.

2. Intensity–to–Time Processing Paradigm

The intensity–to–time processing paradigm implements global operations by aggregating only a few of the input data at a time. Inspired by biological vision [20], the paradigm is based on the notion that stronger input signals elicit responses before weaker ones. Assuming that the inputs have different intensities, the intensity–to–time paradigm separates responses in time allowing a global processor to makes decisions based only on a few inputs at a time. The more time allowed, the more responses are received; thus, the global processor incrementally builds a global decision first based on several, and eventually based on all, the inputs. The key is that some preliminary decision about the environment can be made as soon as the first responses are received. Therefore, this paradigm has an important place in low–latency vision processing.

The architecture supporting intensity–to–time processing is shown in Fig. 1. After a common reset signal at $t=0$, a cell k generates an event at the instant:

$$t_k = f(I_k) \qquad \text{(EQ 1)}$$

where *f(.)* is a monotonic function, and I_k the radiation received by the cell *k*. Therefore, any two cells receiving radiation of different magnitude generate events at different times. If *f(.)* is decaying, then the ordering of events is consistent with a biological system: stronger stimuli elicit responses before weaker ones.

A global processor receives and processes events. In addition, there can be a local processor attached to each cell. The generated events then control the instant when a local processor in each cell performs at least one predetermined (i.e., pre–wired or pre–programmed) operation. By separating the input data in time, the intensity–to–time processing paradigm eases the global data aggregation and computation: 1) the global processor processes only a few events at a time, 2) the communication resources are shared by many cells, and 3) the global processor and local processors infer the input operand intensity by measuring the time an event is received (e.g., $I_k = f^{-1}(t_k)$.)

Traditionally, the intensity–to–time relationship has been used in single and double slope A/D converters [10]. In vision, it has been used to improve diffusion–based image segmentation [7] — a local operation, and for image acquisition in a SIMD architecture [9] — an architecture well suited only for local operations. In contrast, our architecture allows global operations and shares some features of traditional MIMD parallel processing. Namely, the local processors perform their operations *asynchronously*, an essential feature for the quick response and the low latency performance of parallel systems [3].

The intensity–to–time is closely related to the event–address neural communication schemes proposed by a number of researchers [15] [12] [17]. In these schemes the plurality of artificial neurons fire pulses (i.e. *events*) at rates reflecting their individual levels of activity. The goal is to communicate this activity to other neurons or to an output device. The event–address scheme shares communication wires by communicating the identity (i.e. *address*) of the neuron when it fires a pulse. Since the time is inherently measured across the entire system, the receiver recovers the firing rate for each transmitting neuron. The intensity–to–time paradigm synchronizes the responses at the beginning of operations and deals with the time intervals each “neuron” takes to fire its first pulse.

3. Sorting Image Sensor Chip

By using the intensity–to–time paradigm, we have developed a sorting computational sensor — an analog VLSI sensor which sorts the pixels of an input image by their intensities. The chip detects an image focused thereon and computes an *image of indices*. The image of indices has a uniform histogram which has several important properties: 1) the contrast is maximally enhanced, 2) the available dynamic range of readout circuitry is equally utilized, i.e., the values read out from the chip use available bits most efficiently, and 3) the image of indices never saturates, and always preserves the same range (e.g., from 1 to N, where N is the number of pixels). During the computation, the chip computes a cumulative histogram — one global descriptor of the detected scene — and reports it with low–latency on one of the pins. In fact, the global cumulative histogram is used in top–down fashion to update information in local processors and produce the image of indices.

The sorting operation is accomplished in the following way. The plurality of cells detect light and generate events according to (EQ 1), where $f(\ )$ is decaying function. Cells receiving more light generate events before cells receiving less light. The global processor counts the events generated within the array. Since the intensities are ordered in time, the count represents the order, or index, of the cell that is generating an event next. For example, if K events were generated, when a $K+1^{st}$ cell generates the event, the order of the $K+1^{st}$ cell is K. Therefore, the sorting is done by associating the cell(s) currently generating an event with the current count produced by the global counter.

Fig. 2 shows the circuit of the sorting sensor. The global processor/counter comprises an array of constant current sources I_o, a current–to–voltage converter (resistor R), a voltage follower, and wires W_{in} and W_{out}. Upon the arrival of the event generated by a cell, the corresponding individual current source I_o is turned on via switch S_6. The current sources are summed in the wire W_{out}. The cumulative current in the wire W_{out} continuously reports the number of cells that have responded with an event—the global count. The cumulative current is converted to a voltage via resistor R, and fed in a top–down fashion to the local processors in the cell array via wire W_{in}.

The local processor in each cell comprises a track–and–hold (T/H) circuit. The T/H tracks the voltage on W_{in} until the event in the cell is generated. At that point the local processors remember voltage supplied by the global counter on the wire W_{in}. Thus, the appropriate index is assigned to each cell.

The remaining portion of the cell comprises the photo sensitive intensity–to–time event generator which generates an event according to (EQ 1). Fig. 3 shows the simulation of the circuit operation for the sorting sensor with four cells. A photodiode *PD* operating in the photon flux integrating mode [21] detects the light. In this mode of operation the capacitance of the diode is charged to a high potential and left to float. Since the diode capacitance is discharged by the photo current, the voltage decreases approximately linearly at a rate proportional to the amount of light impinging on the diode (Fig. 3, top graph).

The diode voltage is monitored by a CMOS inverter. Once the diode voltage falls to the threshold of the inverter, the inverter’s output changes state from low to high (Fig. 3, second graph). A switch S_3 is included to provide a positive feed back and force rapid latching action. The transition in the output of the inverter represents the event generated by the cell. It controls the *instant* when the capacitor C in the T/H memorizes the voltage supplied on the wire W_{in}. It also controls the instant when the current I_o is supplied to the wire W_{out}.

The voltage on the wire W_{in} (Fig. 3, third graph) represents the index of a cell that is changing state and is supplied to the global input wire. The T/H within each cell follows this voltage until it is disconnected, at which point a capacitor C retains the index of the cell (Fig. 3, bottom graph). The bottom graph shows that the cell with the highest intensity input has received the highest “index,” the next cell one “index” lower, and so on. The charge from the capacitors is readout by scanning the array after all the cells responded or after a predetermined time-out.

The sorting sensor computes several important properties about the image focused thereon. First, the time when a cell k triggers is approximately inversely proportional to the input radiation it receives:

$$t_k = C\frac{(V_{DD} - V_{th})}{I_k + I_d} \tag{EQ 2}$$

where C is the diodes capacitance, V_{DD} the power supply voltage, V_{th} the threshold voltage of the inverter, and I_k photocurrent approximately proportional to the radiation, and I_d is the dark current.

Second, by summing up the currents I_o, the global processor knows at each given time how many cells have responded with an event. Since events are generated according to (EQ 1), the cumulative current in the wire W_{out}, or its inverse, the voltage on the wire W_{in}, are recognized as being temporal representations of the cumulative histogram of the input data, with the horizontal axis being proportional to $1/t$. The time derivative of the cumulative histogram signal is related to a histogram of the input image [2]. The cumulative histogram is one global property of the scene that is reported by the chip with very low latency. Such information can be used for preliminary decision making as soon as the first responses are received. In fact, it is used on-chip to quickly adapt index values for every new frame of input data.

4. VLSI Realization and Evaluation

A 21 x 26 cell sorting sensor has been built in 2μ CMOS technology. The size of each cell is 76μ by 90μ, with 13% fill factor. The micrograph of the chip is shown in Fig. 4. An image was focused directly onto the silicon. The cumulative histogram waveform as well as the indices from the sorting sensor were digitized with 12–bit resolution.

Scene *1* of an office environment was imaged by the sorting chip under common office illumination coming from the ceiling. Fig. 5(a) and (b) shows the cumulative histogram of the *scene* and the image of indices both computed by the chip. We evaluated the histogram of the indices which is shown in Fig. 5(c). From Fig. 5(c) it is seen that most pixels appeared to have different input intensities and, therefore, received different indices. Occasionally, as many as 3 pixels were assigned the same index. Overall, the histogram of indices is uniform, indicating that the sorting chip has performed correctly.

Scene *2* from the same office was also imaged. Scene 1 (Fig. 5) contains more dark regions than Scene 2 (Fig. 6) because the moderately bright wall in the background is replaced by the dark regions of the person in partial shadow. Therefore, the chip takes longer to compute Scene 1 than Scene 2, but the dynamic range of the output indices is maintained. The total time shown on the time sample axis of the cumulative histograms is about *200ms.*

By producing the cumulative histogram waveform and the image of indices, the sorting computational sensor provides all the necessary information for the inverse mapping — the mapping from the indices to the input intensities. Fig. 7(a) shows the image of indices for Scene 1 and the image of inferred input intensities. Fig. 7(b) includes an image taken by a commercial CCD camera for showing natural light conditions in the office environment from which Scene 1 was taken. The inferred input intensities closely resemble the natural low contrast conditions in the environment.

There is a total of 546 pixels in the prototype described in this paper. The uniform histogram of indices (Fig. 5c and Fig. 6c) indicates that most of the pixels received different indices. Therefore, without special considerations as to the illumination conditions, low–noise circuit design and temperature and dark current control, our lab prototype readily provides indices with more than 9 bits of resolution. Furthermore, the range of indices remains unchanged (from 0 to 545) and the indices maintain uniform histogram regardless of the range of input light intensity or its histogram.

5. Sorting Sensor Image Processing

The data that are stored in the local processors are provided by the global processor. These global data — a function of time — define a mapping from the input intensities to output data. For the sorting operation, this global function is the cumulative histogram computed by the chip itself. In general, when appropriately defined, this global function enables the sorting sensor to perform numerous other operations/mappings on input images. Examples of such operations include histogram computation and equalization, arbitrary point-to-point mapping, region segmentation and adaptive dynamic range imaging. In fact, in its native mode of operation — sorting — the chip provides all the information necessary to perform any mapping during the readout.

Histogram Equalization. When the voltage of the cumulative histogram (computed by the chip itself) is supplied to the local processors, the generated image is a histogram–equalized version of the input image [2]. This is the basic mode of operation for the sorting chip and has been illustrated in the previous section.

Linear Imaging. When the waveform supplied to the input wire is inversely proportional to time, the values stored in the capacitors are proportional to the input intensity, implementing a linear camera. The results of such mapping have been illustrated in Fig. 7. As expected, the result is similar to the image obtained by the linear CCD imager. (The CCD image and sorting sensor image are obtained within minutes from each other, under the same illumination condition.)

Scene Change Detection. Analyzing the change in the histogram pattern is a basic technique to classify images or detect a scene change. The sorting computational sensor computes the cumulative histogram at real–time and can be used for low-latency scene discrimination/surveillance without requiring the image to be read out. For example, by comparing the cumulative histograms for Scenes 1 and 2, one could conclude that the brightest pixels (i.e. computer monitor) did not change (see Fig. 8). One also could conclude that the remainder of the image in Scene 2 is brighter than in Scene 1, since the Scene 2 takes less time to compute. Other more intelligent and detailed reasoning about the scene based only on the cumulative histogram is possible.

Image Segmentation. Thresholding is a rudimentary technique to segment an image into regions. The cumulative histogram can be used to determine this threshold. Pixels from a single region often have pixels of similar intensity that appear as clusters in the image histogram [2]. The values which ought to be stored in the cells can be generated to correspond to the “label” of each such region. A global processor can be devised that performs this labeling by updating the supplied value (i.e. label) when the transition between the clusters in the (cumulative) histogram is detected. An example of segmentation is shown in Fig. 10(b) and Fig. 10(c) in which the illuminated and shadowed regions respectively are “colored” as a black region.

Adaptive Dynamic Range Imaging. For faithful imaging of scenes with strong shadows, a huge dynamic range linear camera is needed. For example, the illumination of the scene which is directly exposed to the sunlight is several orders of magnitude greater than the illumination for the surfaces in the shadow. Due to the inherently large dynamic range of the sorting sensor, both illuminated and shadowed pixels can be mapped to the same output range during a single frame.

We demonstrate this concept with back illuminated objects. Fig. 9 shows a global view of this scene as captured by a conventional CCD camera. Due to the limited dynamic range of the CCD camera, the foreground is poorly imaged and is mostly black. (The white box roughly marks the field–of–view for the sorting sensor.)

When the scene is imaged with the sorting sensor (Fig. 10a), the detail in the dark foreground is resolved, as well as the detail in the bright background. Since all 546 indices are competing to be displayed within the 256 levels allowed for the postscript images in this paper, one enhancement for purpose of human viewing is to segment the image and amplify only dark pixels. The result is shown in Fig. 10b. Conversely, as shown in Fig. 10c, the bright pixels can be spanned to the full (8 bit) output range. Finally, if these two mappings are performed simultaneously, the shadows are removed (Fig. 10d.)

The same method can be applied to the image obtained from a standard CCD camera. If the CCD image of Fig. 9 is cropped to the white box, and such an image is histogram–equalized, we arrive at the result shown in Fig. 11a. This image is analogous to the image of indices obtained by the sorting sensor (Fig. 10a). Due to the limited dynamic range, noise and quantization, the CCD image only resolves the face with 2–3 bits. The histogram–equalized image from the CCD is used for further mapping using the same steps as for Fig. 10d. Due to obvious reasons, the result is poor. In contrast, the sorting computational sensor allocates as many output levels (i.e., indices) as there are pixels within the dark region, or the entire image for that matter. By comparing Fig. 10(d) and Fig. 11(b), the superior utilization of the sensory signal with the sorting chip is obvious.

The adaptation of the dynamic range of the sorting sensor is also illustrated in Fig. 12, showing a sequence of 93 images of indices computed by the sorting sensor. The sensor was stationary, and the only changes in the scene are due to subject movement. By observing the wall in the background, we can see the effects of adaptive dynamic range: even though the physical wall does not change the brightness, it appears dimmer in those frames in which bright levels are taken by pixels which are physically brighter (e.g., subject's face and arm). When the subject turns and fills the field–of–view with dark objects (e.g., hair) the wall appears brighter since it is now taking higher indices. Also, note that the maximum contrast is maintained in all the images since all images of indices have a uniform histogram.

6. Error Analysis

Theoretically, the dynamic range of the scene detectable by the sorting sensors is unlimited. Of course, in practice the actual dynamic range of the sensor will be determined by the capabilities of the photodetector, as well as by the switching speed and dark current levels.

First we investigate the mismatch of the cells. Even when receiving same light levels, the cells do not respond at the same time. This determines the fundamental accuracy of the intensity–to–time paradigm. Given (EQ 2), the input photocurrent I_λ can be found as:

$$I_\lambda = \frac{CV}{t} - I_d \qquad \text{(EQ 3)}$$

where $V = V_{dd} - V_{th}$, and I_d is the dark current. The relative error can be found as:

$$\left(\frac{\sigma_{I_\lambda}}{I}\right)^2 = \left(\frac{\sigma_{I_d}}{I}\right)^2 + \left(\frac{\sigma_C}{C}\right)^2 + \left(\frac{\sigma_V}{V}\right)^2 + \left(\frac{\sigma_t}{t}\right)^2 \qquad \text{(EQ 4)}$$

where $I = I_\lambda + I_d$, σ_{Id} represents fluctuation of the dark current over the sensor area, σ_C represents fluctuations of the photodetector capacitance (e.g., mismatch of the photodetectors), σ_V represents the mismatch of the threshold voltages and the diode's reset noise, and σ_t represents the fluctuation in the switching speed of the control element. After substituting (EQ 2) in the last term in (EQ 4) relative error becomes:

$$\left(\frac{\sigma_{I_\lambda}}{I}\right)^2 = \left(\frac{\sigma_{I_d}}{I}\right)^2 + \left(\frac{\sigma_C}{C}\right)^2 + \left(\frac{\sigma_V}{V}\right)^2 + \left(\frac{\sigma_t}{CV}I\right)^2 \qquad \text{(EQ 5)}$$

$$\left(\frac{\sigma_{I_\lambda}}{I}\right)^2 = \frac{A^2}{I^2} + B^2 + C^2 I^2 \qquad \text{(EQ 6)}$$

where A, B, and C substitute constant terms in (EQ 5). This error model follows the intuition: for high levels of illumination, when the cells respond quickly, the dominant cause of error is the fluctuation in the switching speed; for low illumination levels, the dominant factor is the fluctuation in the dark current.

The constants A, B and C were experimentally determined from the prototype chip. Without the lens in front of the sensor, the sensor was illuminated by a halogen light source reflected from a white cardboard. As the cardboard was positioned several meters from the sensor, the illumination field was considered uniform over the sensor's surface. The amount of light falling on the sensor's surface was controlled by changing the angle between the light source and the cardboard. The cumulative histogram waveform was gathered for 43 different light levels. (We don't know the absolute value of light levels. As an illustration for a reader, the brightest level in our experiment was comparable to the level of an average size room (e.g., 12' x 12') illuminated with a 150W bulb; the darkest level was comparable to the same room illuminated with a desk lamp.) From the cumulative histogram waveforms and (EQ 3), the mean value *I*, and standard deviation σ_I, were computed in arbitrary current units [ACU]. (1 ACU = 1/s, i.e., 1 ACU triggers an event according to (EQ 2) 1 second after the beginning of the frame integration.) The error model (EQ 6) was fitted to the data. The results are tabulated in Table 1, and graphed in Fig. 13.

TABLE 1. Error performance of the sorting sensor prototype.

A	2.2860e-2 [ACU]	
B	5.0636e-3	
C	4.2862e-5 [1/ACU]	
$\frac{\sigma_I}{I} = 1$	Imin	0.0229[acu]
	Imax	23330[acu]
	Dyn. range	1:1020565
$\frac{3\sigma_I}{I} = 1$	Imin	0.0686[acu]
	Imax	7776 [acu]
	Dyn. range	1:113373

For the signal–to–noise ratio (SNR) of one, the dynamic range of the sensor based on the model is over 10^6. If the three sigma rule is used for the noise limits, the dynamic range is over 10^5. However, the detectable lower limit on the input photocurrent is determined by the level of the dark current. In the experiment, we determined that the average dark current is about 0.2 ACU; therefore, for SNR=1 we require the lowest input photocurrent to be 0.2 ACU. Then, the dynamic range is 1: 116650 for one sigma rule and 1:38880 for the three–sigma rule. Given constant dark current, the dynamic range is limited by the error the sensor makes when detecting the high illumination levels. The dominant source for this error is the fluctuation in the turn–on time of the inverters. In our experiment, this fluctuation is about 43µs (i.e., constant C). This is very high switching fluctuation. It is probably due to the fact that 1) the input voltage is slowly approaching threshold level of the inverter, thus causing the long transition times at the inverter's output, 2) the positive feedback transistor is active only after the "decision" to trip is made, 3) in a static CMOS inverter the p– and n–channel transistors "fight" each other for slow–changing inputs, and 4) there could be some systematic limitation in our instrumentation setup and/or the conditions under which we assume equal illumination for all pixels. In all, the switching fluctuation is approximately10% of inverter output transition time (i.e., rise time) in the cell receiving the highest intensity in our experiment, which is reasonable. A higher gain thresholding element would probably perform better. This hypothesis will be verified with a new prototype currently being fabricated. Other sources of error, fluctuations in the dark current (i.e., constant A) and mismatch of C and V (i.e., constant B), are within reasonable limits. Relative error for the dark current is approximately 10%, while the lumped relative error for C and V is approximately 0.5%.

The second issue we would like to consider is the error the sorting sensor makes when computing the cumulative histogram. This error is due to the mismatch of the current sources I_o. Since there are typically thousands of cells in the sorting image sensors, the level of current I_o is very low, pushing the corresponding transistors into the subthreshold regime. In this regime, the current sources could mismatch by 100%, i.e., one current source can be twice as large as another [14]. Nonetheless, the monotonousness in the cumulative histogram is maintained. When the cumulative histogram is used for inverse mapping, the mapping from indices to the input intensities, the error in cumulative histogram is not significant as it will be directly undone.

The error that could be significant when mapping from indices to input intensities, however, is the readout error for each index. If the scene produced long horizontal segments in the cumulative histogram, such as the example in Fig. 10a, then a small error in index can result in a large error in inferred response time for a particular cell. This problem can be handled by prohibiting the mapping process to return times within the interval of the long horizontal segments in the cumulative histogram. A few pixels may be grossly misclassified, but overall recovery of input intensities is good.

7. Conclusion

The intensity–to–time processing paradigm enables VLSI computational sensors to be massively–parallel computational engines which make global computation or overall decisions about the sensed scene and reports such decisions on a few output pins of the chip with low latency. The power of this paradigm is demonstrated with an analog VLSI implementation of sorting — an operation still challenging in computer science when performed on large groups of data. This work shows that, if an appropriate relationship is maintained between the circuitry, algorithm and application, a surprisingly powerful performance can be achieved in a fairly simple but fairly high resolution VLSI vision computational sensor.

Acknowledgments

This research has been partially funded by the ONR Grant N00014–95–1–0591 and by the NSF, Grant MIP–9305494. The authors also acknowledge the critical and constructive comments by the reviewers.

References

[1] J. Aloimonos (edt.), "Special Issue on Purposive, Qualitative, Active Vision," *CVGIP: Image Understanding*, Vol. 56, No. 1, 1992.

[2] D.H. Ballard and C.M. Brown, *Computer Vision*, Prentice-Hall, 1982.

[3] V. Brajovic, *Computational Sensors for Global Operations in Vision*, Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1996.

[4] 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*, Minneapolis, MN, April 1996, pp. 1638–1643.

[5] V. Brajovic and T. Kanade, "Computational Sensors for Global Operations", *IUS Proceedings*, pp. 621-630, 1994.

[6] V. Brajovic and T. Kanade, "Computational Sensor for Visual Tracking with Attention," *IEEE Jour. of Solid-State Circuits*, Vol 33, No 8, August 1998.

[7] P. Y. Burgi and T. Pun, "Asynchrony in Image Analysis: using the luminance–to–response–latency relationship to improve segmentation," *J. Opt. Soc. Am. A*, Vol. 11, No. 6, June 1994, pp. 1720–1726.

[8] S. P. DeWeerth, "Analog VLSI Circuits for Stimulus Localization and Centroid Computation," *Intl. Jour. of Comp. Vision*, Vol. 8, No. 3, 1992, pp. 191-202.

[9] R. Forchheimer and A. Astrom, "Near–Sensor Image Processing: A New Paradigm," I*EEE Trans. on Image Proc.*, Vol. 3, No. 6, pp. 736–746, November 1994.

[10] R. I. Geiger, L. P. E. Allen, N.R. Strader: *VLSI Design Techniques for Analog and Digital Circuits*, McGraw–Hill, 1990.

[11] T. Kanade and R. Bajcsy, "Computational Sensors: A Report from DARPA workshop," *Image Understanding Workshop Proceedings*, 1993.

[12] J. Lazzaro, J. Wawrzynek, M. Mahowald, M. Sivilotti, D. Gillespie: "Silicon auditory processors as computer peripherals," *IEEE Transactions on Neural Networks*, Vol 4, No 3, May 1993

[13] B. Mathur and C. Koch, ed. *Visual Information Processing: From Neurons to Chips*, *Proc. SPIE*, Vol. 1473, 1991.

[14] C. Mead, *Analog VLSI and Neural Systems*. Reading, MA: Addison-Wesley, 1989.

[15] M. Mahowald, *Computation and Neural Systems*, Ph.D. Thesis, California Institute of Technology, 1992.

[16] T.G. Morris and S.P. DeWeerth, *Analog VLSI Circuits for Covert Attentional Shifts*, MicroNeuro ‘96, Lausanne, Switzerland.

[17] A. Mortara, E.A. Vittoz, P. Venier, “A communication scheme for analog VLSI perceptive systems,” *IEEE Jour. of Solid–State Circuits*, Vol 30, No 6, pp. 660-669, June 1995.

[18] D. Standley, “An Object Position and Orientation IC with Embedded Imager,” *IEEE Journal of Solid-State Circuits*, Vol. 26, No. 12, 1991, pp. 1853-1860.

[19] B. Zavidovique and T. Bernard, “Generic Functions for On–Chip Vision,” *ICPR, Conference D*, The Hague, The Netherlands, 1992. pp. 1–10.

[20] H. Davson (ed.), *The Eye*, Vol. 2A, Academic Press, 1976, chapter by Ripps, H. and R.A. Weale, “Temporal Analysis and Resolution,” pp. 185-217.

[21] Weckler, G.P. “Operation of p-n Junction Photodetectors in a Photon Flux Integrating Mode,” *IEEE Jour. of Solid–State Circuits*, pp. 65–73, Vol. sc–2, No. 3, September, 1967

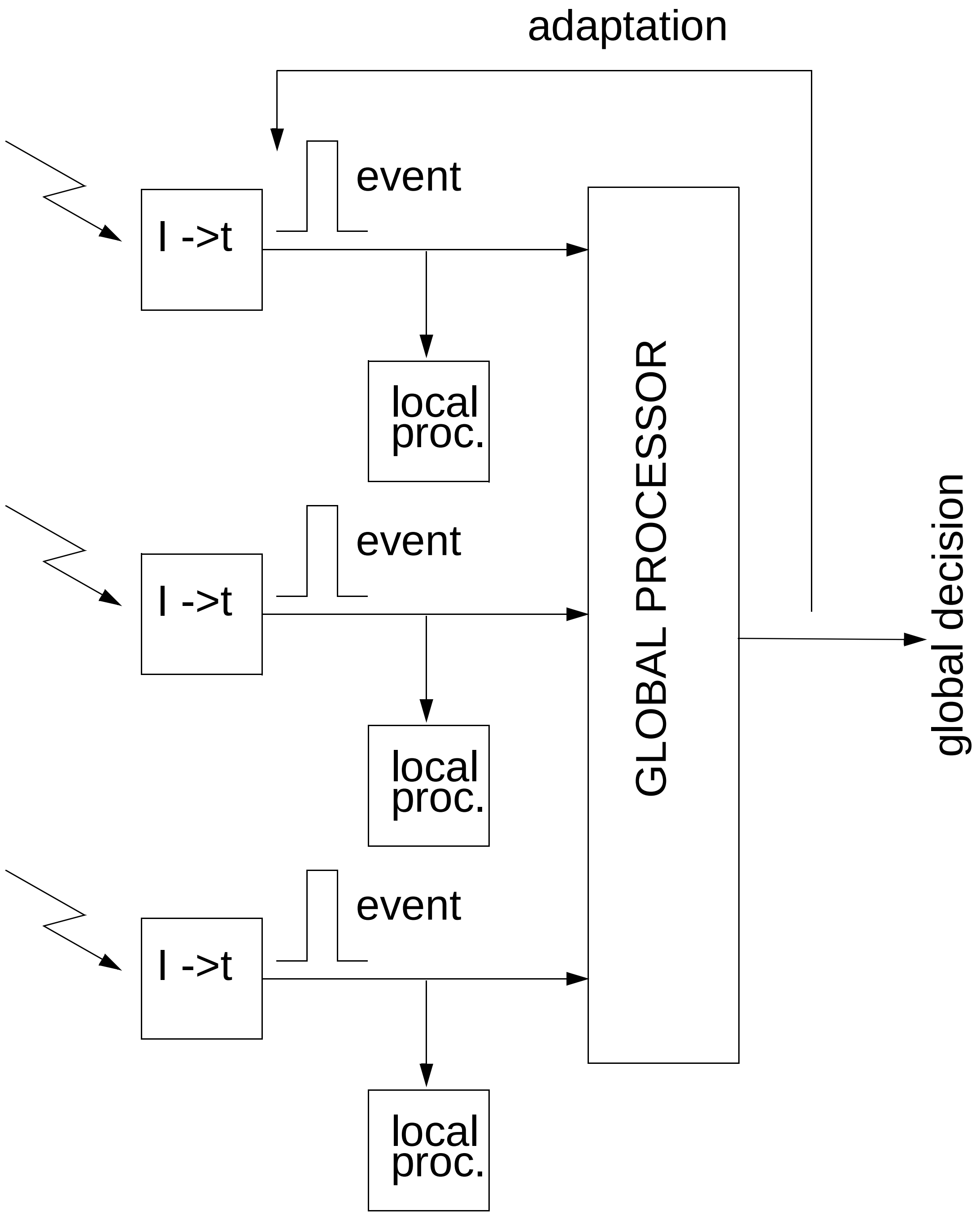


Fig. 1. A computational sensor architecture for the intensity–to–time processing paradigm.

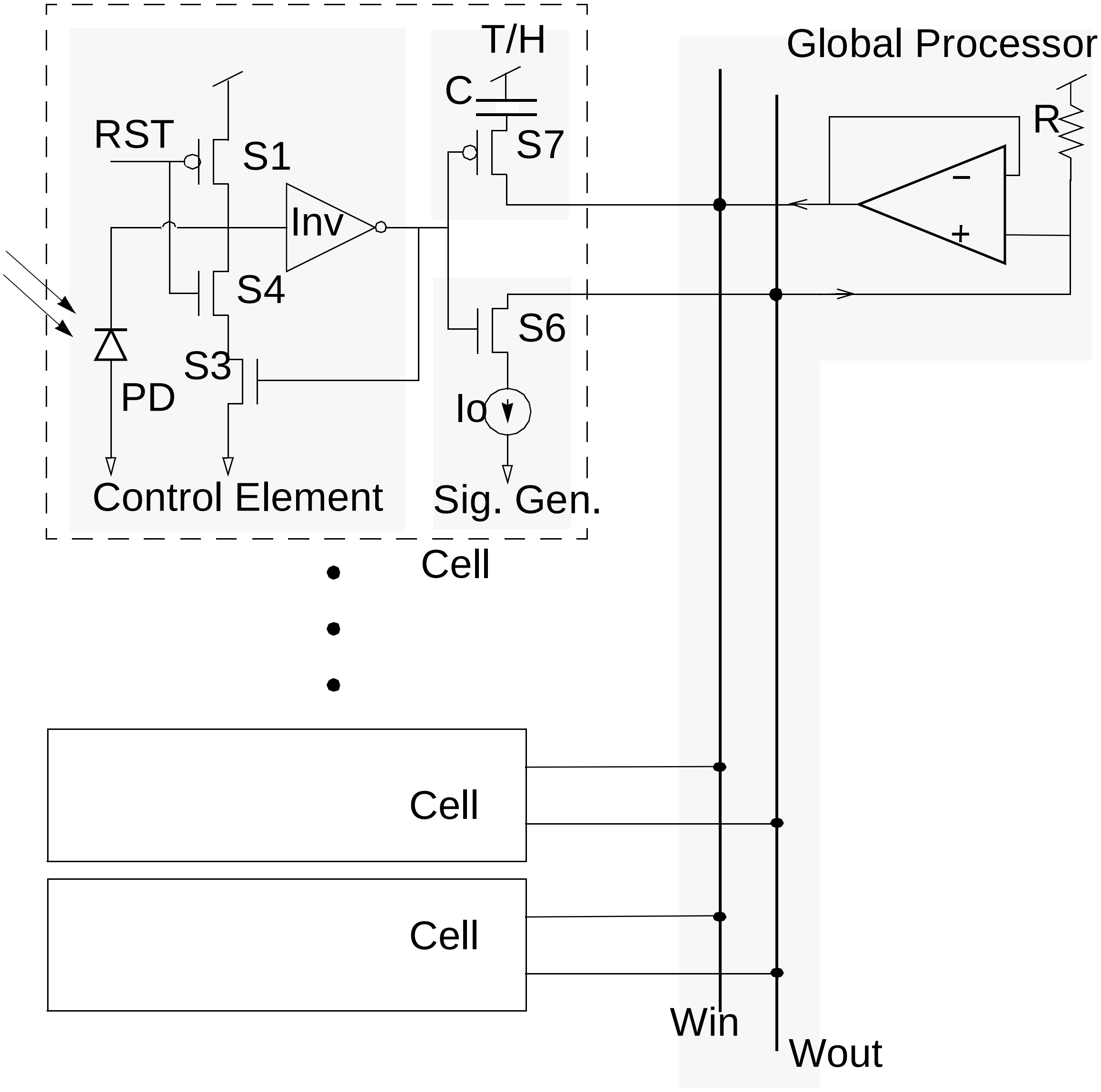


Fig. 2. Schematic diagram of the sorting computational sensor.

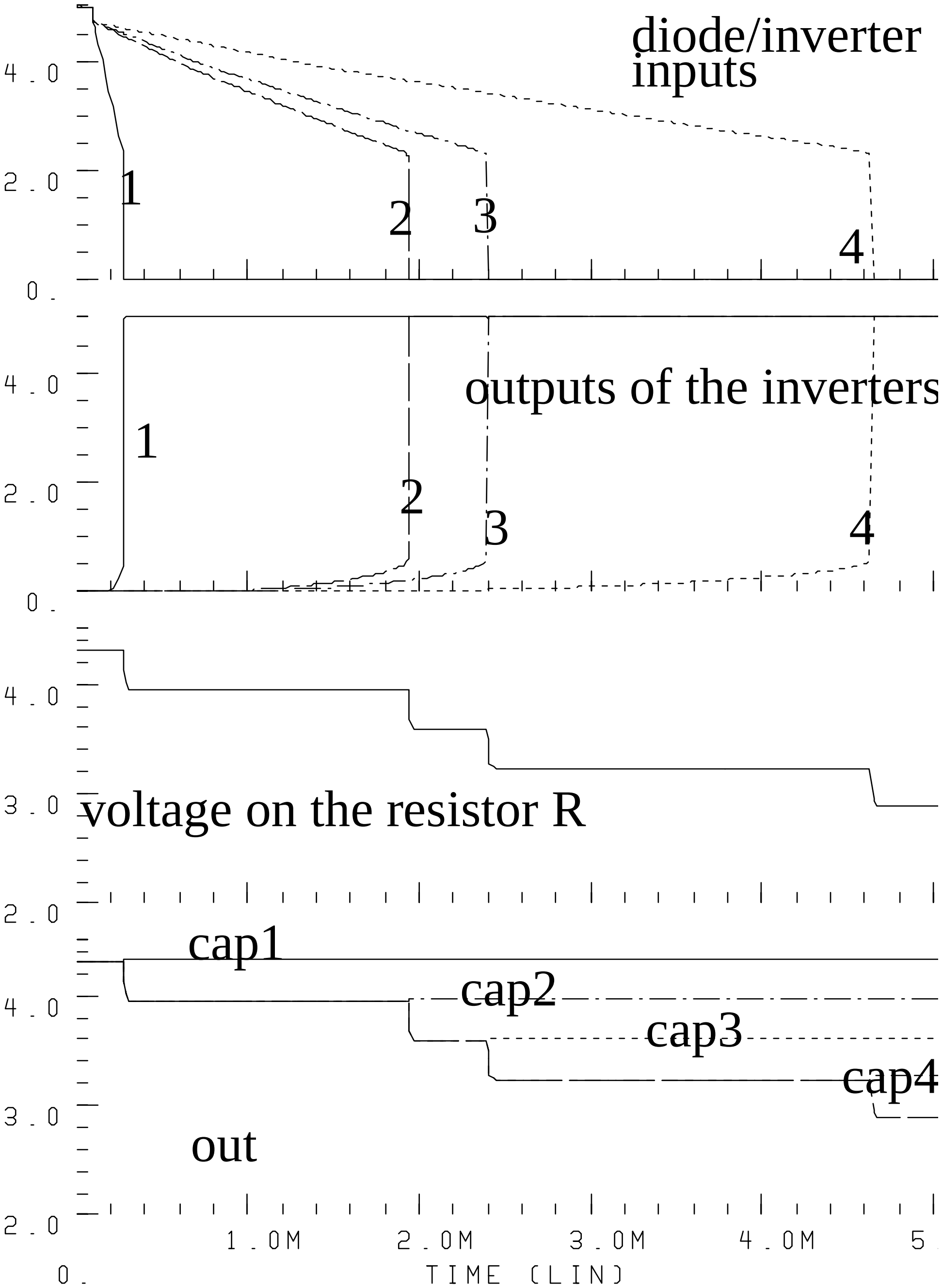


Fig. 3. Sorting computational sensor: a four cell simulated operation.

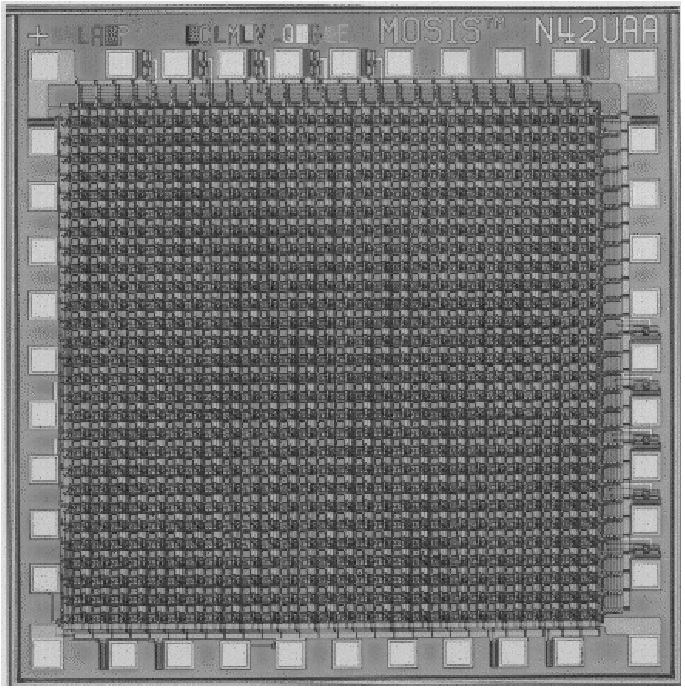
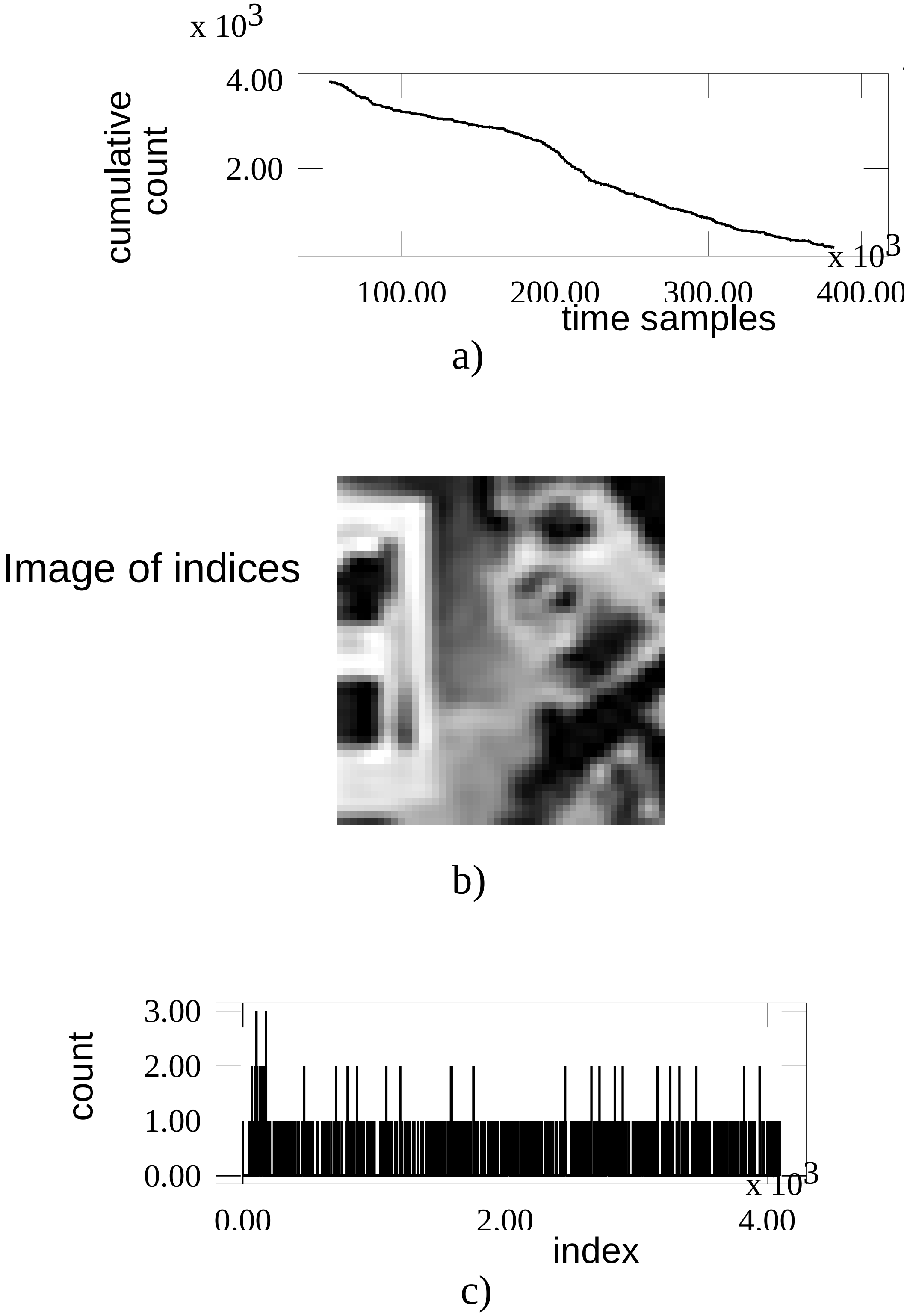


Fig. 4. Micrograph of the sorting chip.



a)

b)

c)

Fig. 5. Scene 1 imaged by the sorting sensor: a) cumulative histogram computed by the chip (voltage on Win), b) image of indices, c) histogram of indices.

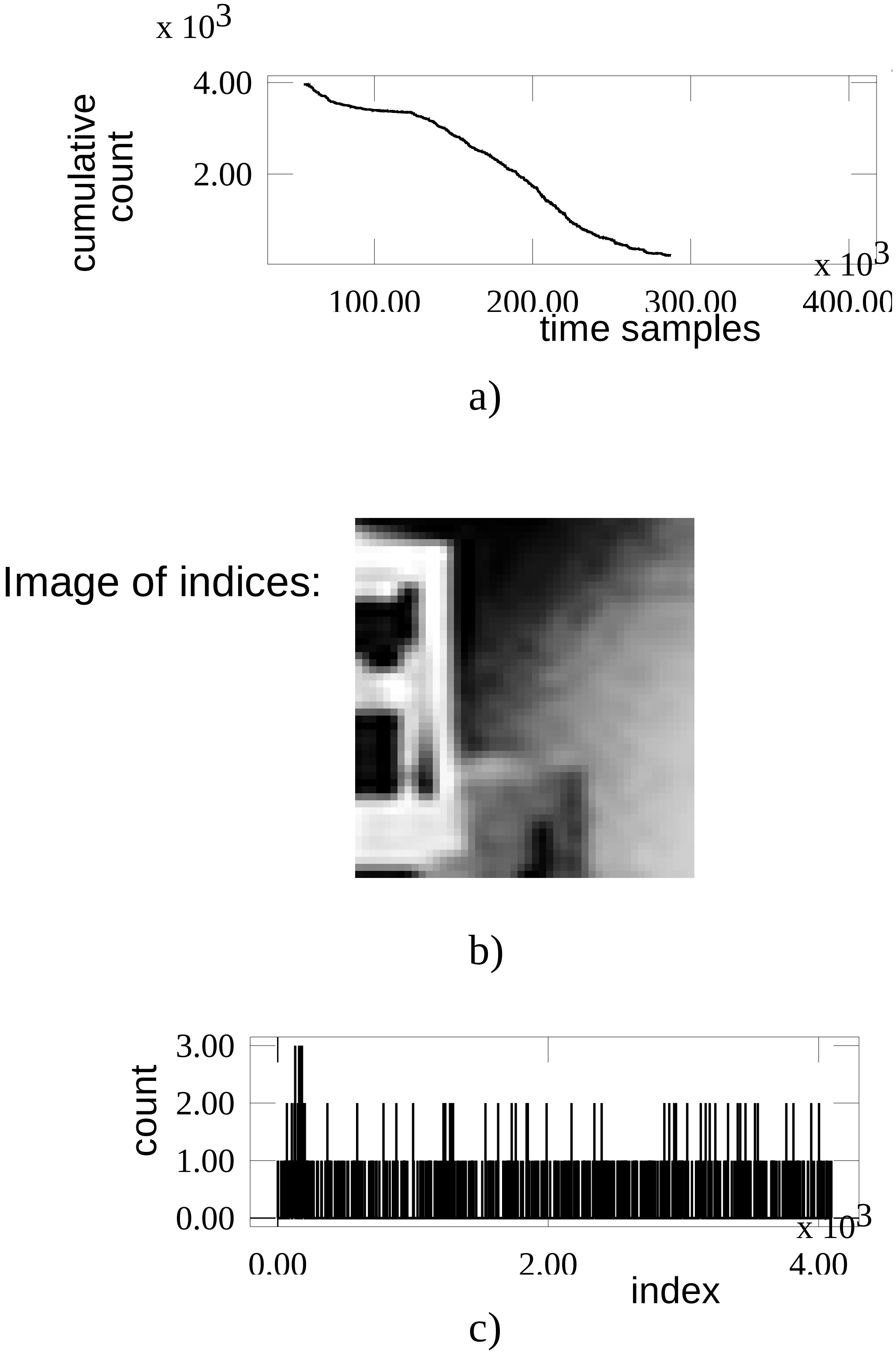


Fig. 6. Scene 1 imaged by the sorting sensor: a) cumulative histogram computed by the chip (voltage on Win), b) image of indices, c) histogram of indices.

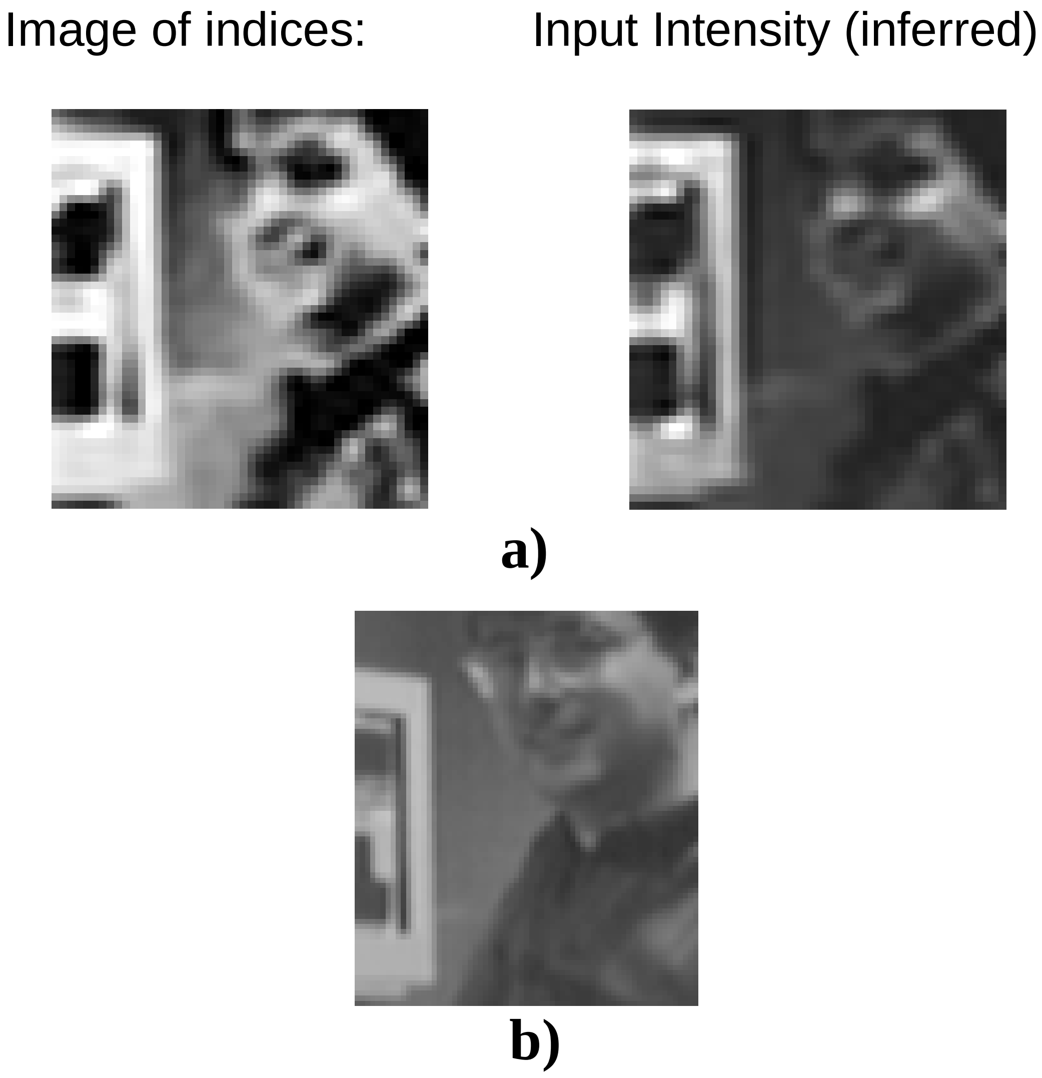


Fig. 7. a) Indices from the sorting sensor and inferred input intensity, b) CCD camera image.

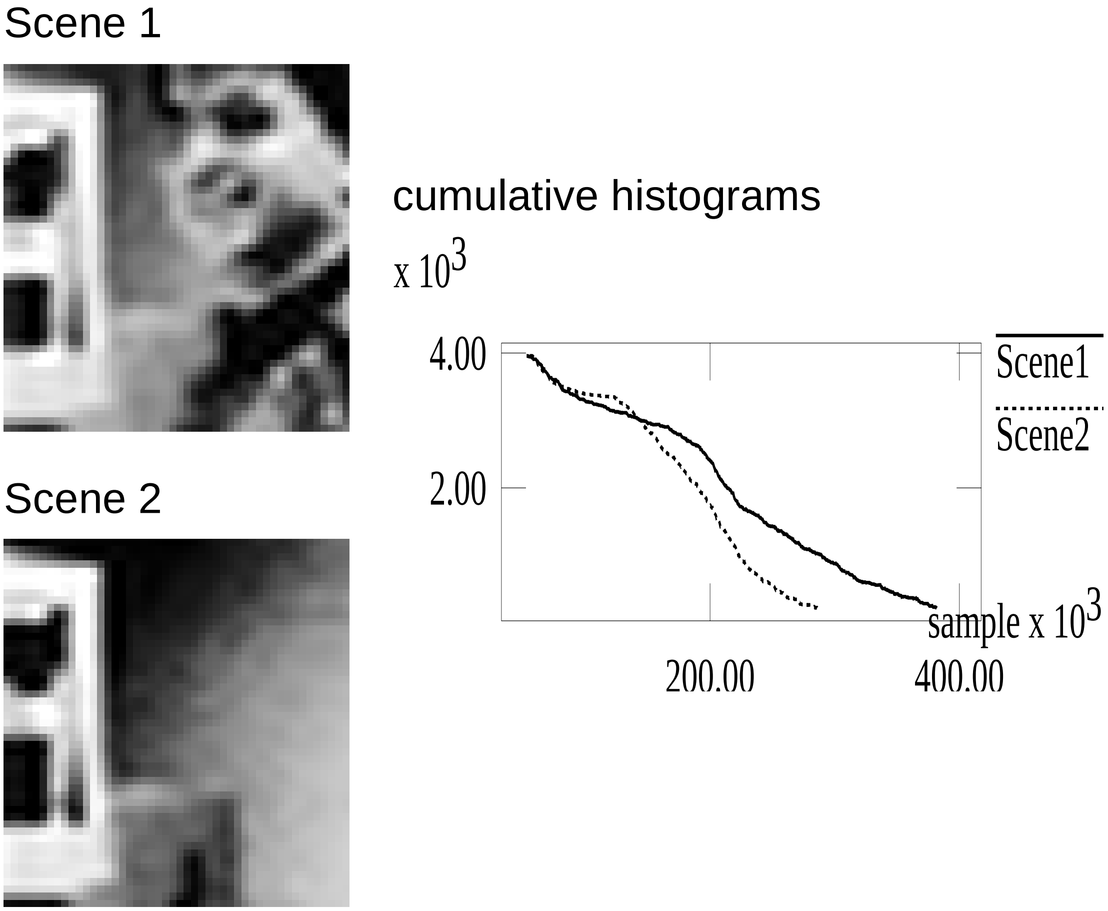


Fig. 8. Detecting a scene change by observing cumulative histograms only.

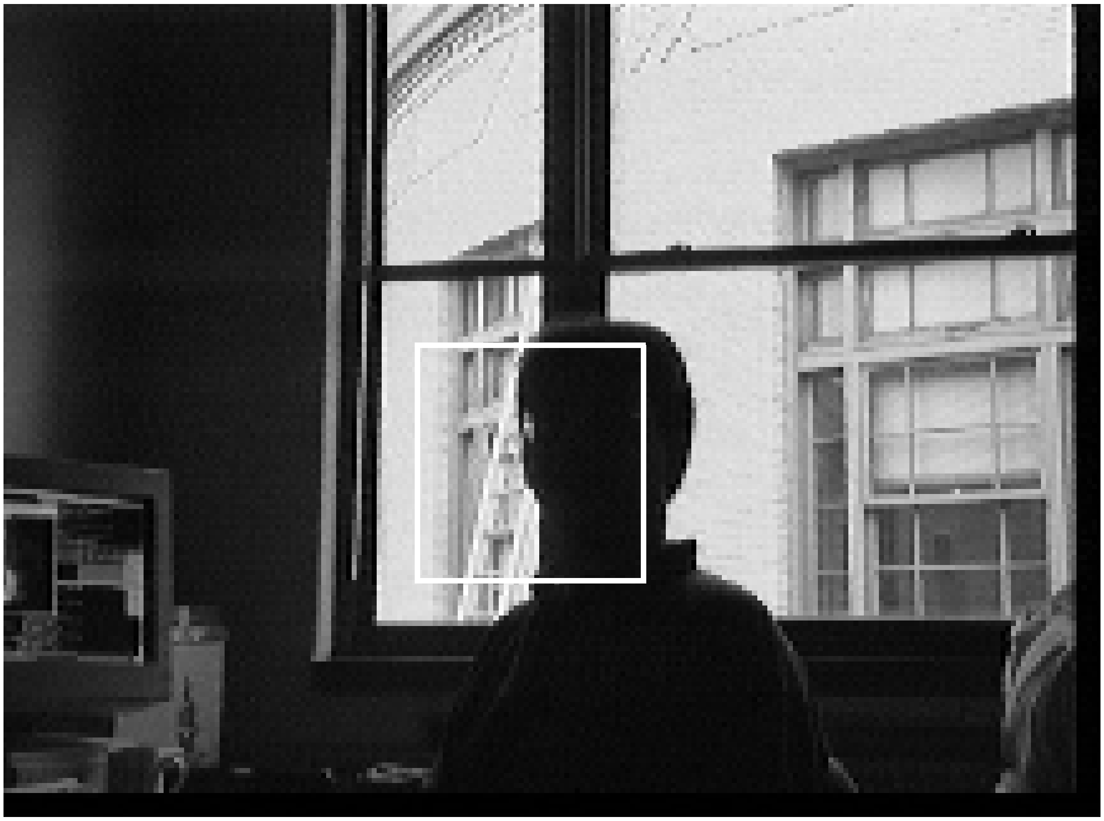
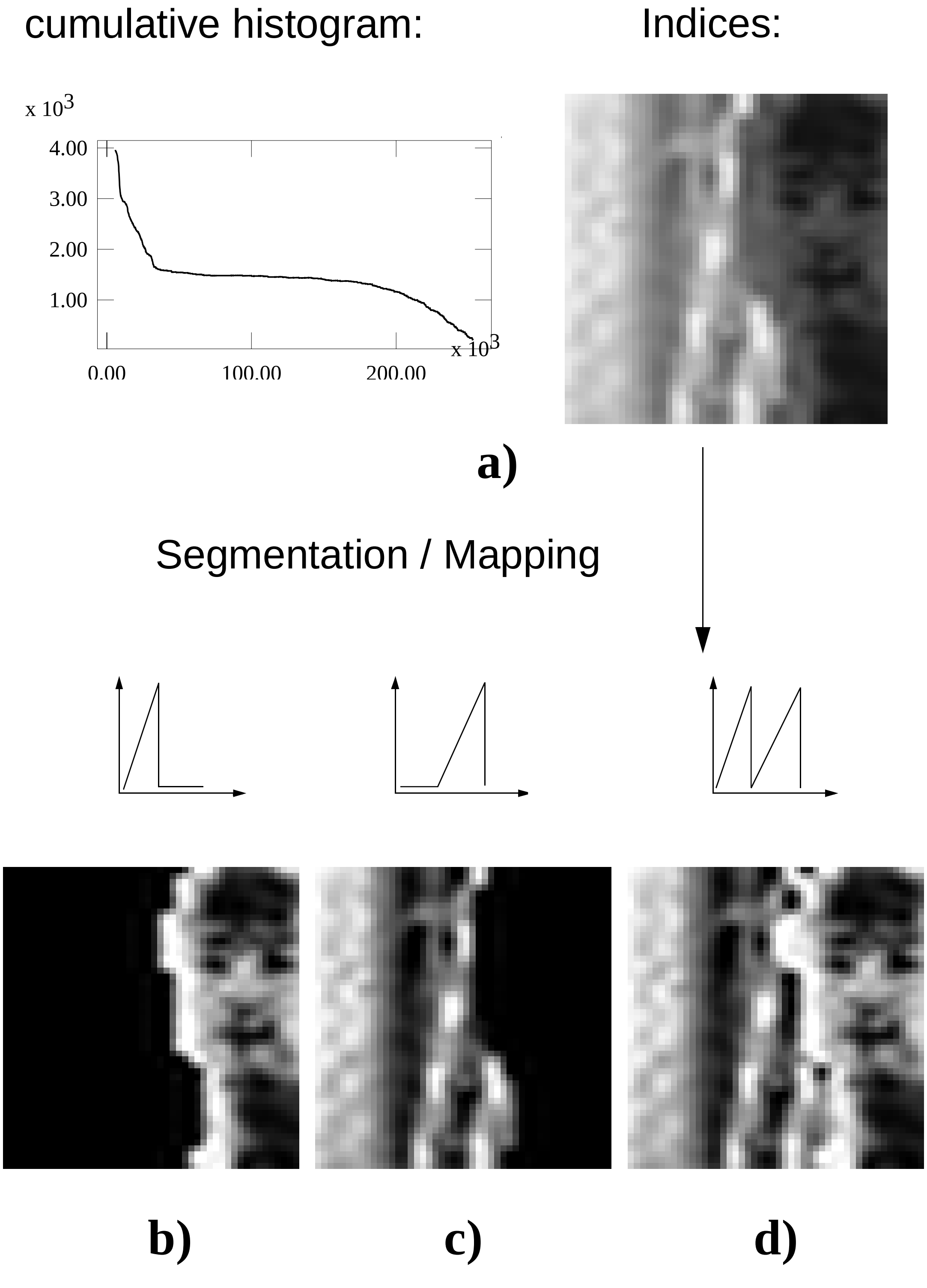


Fig. 9. A scene with backlit objects as captured by a conventional CCD camera.



g. 10. Sorting sensor processing: a) data from the sensors; b) segmentation (viewing the shadowed gion); c) segmentation (viewing illuminated region); d) segmentation and shadow removal.

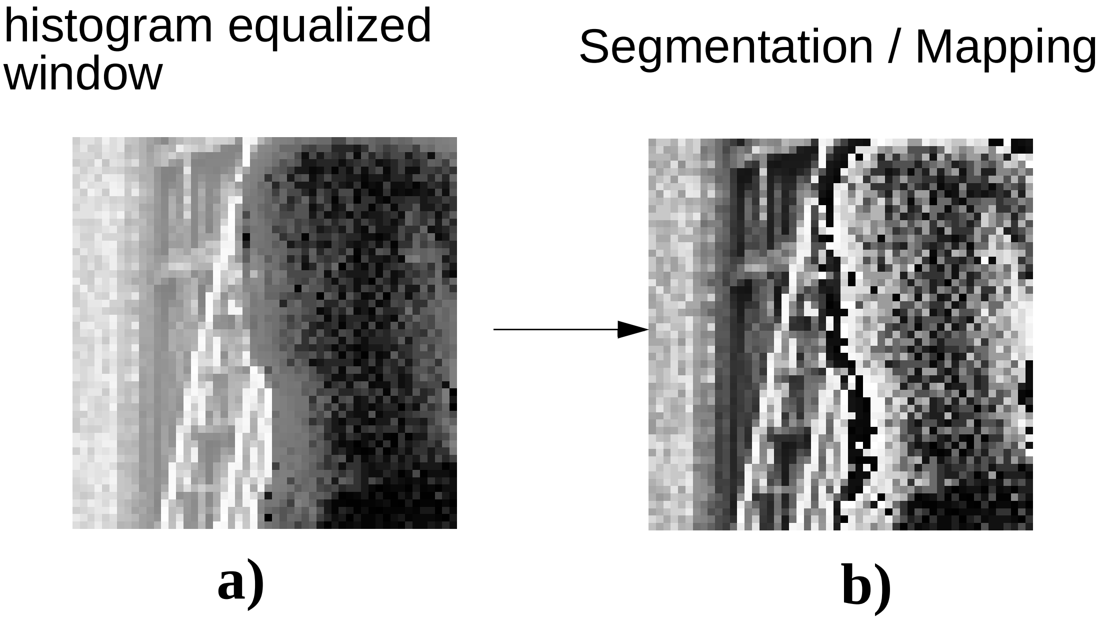


Fig. 11. Conventional CCD camera processing: a) histogram equalization of the window; b) segmentation and shadow removal.

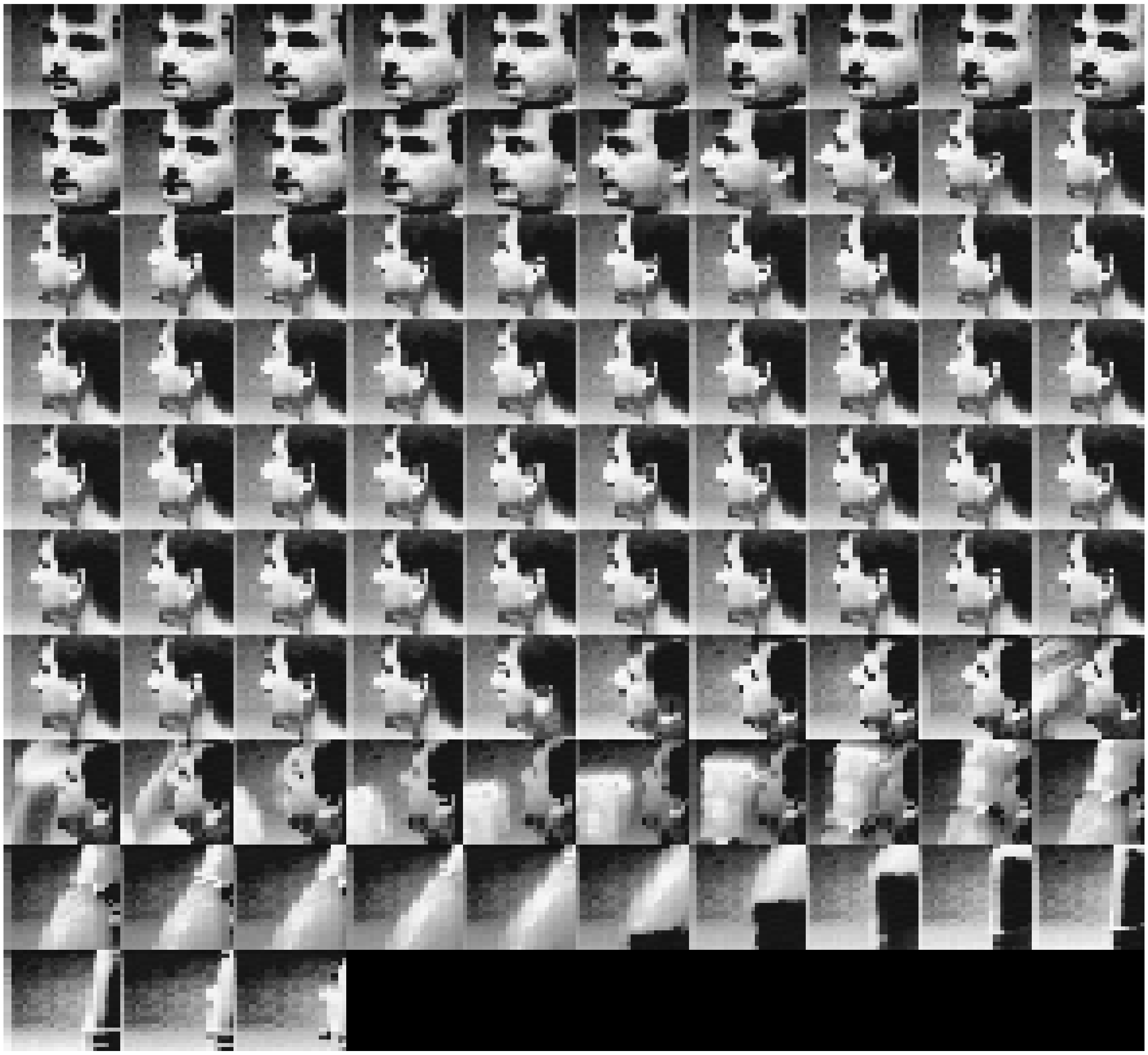


Fig. 12. Sequence of images of indices computed by the sorting sensors.

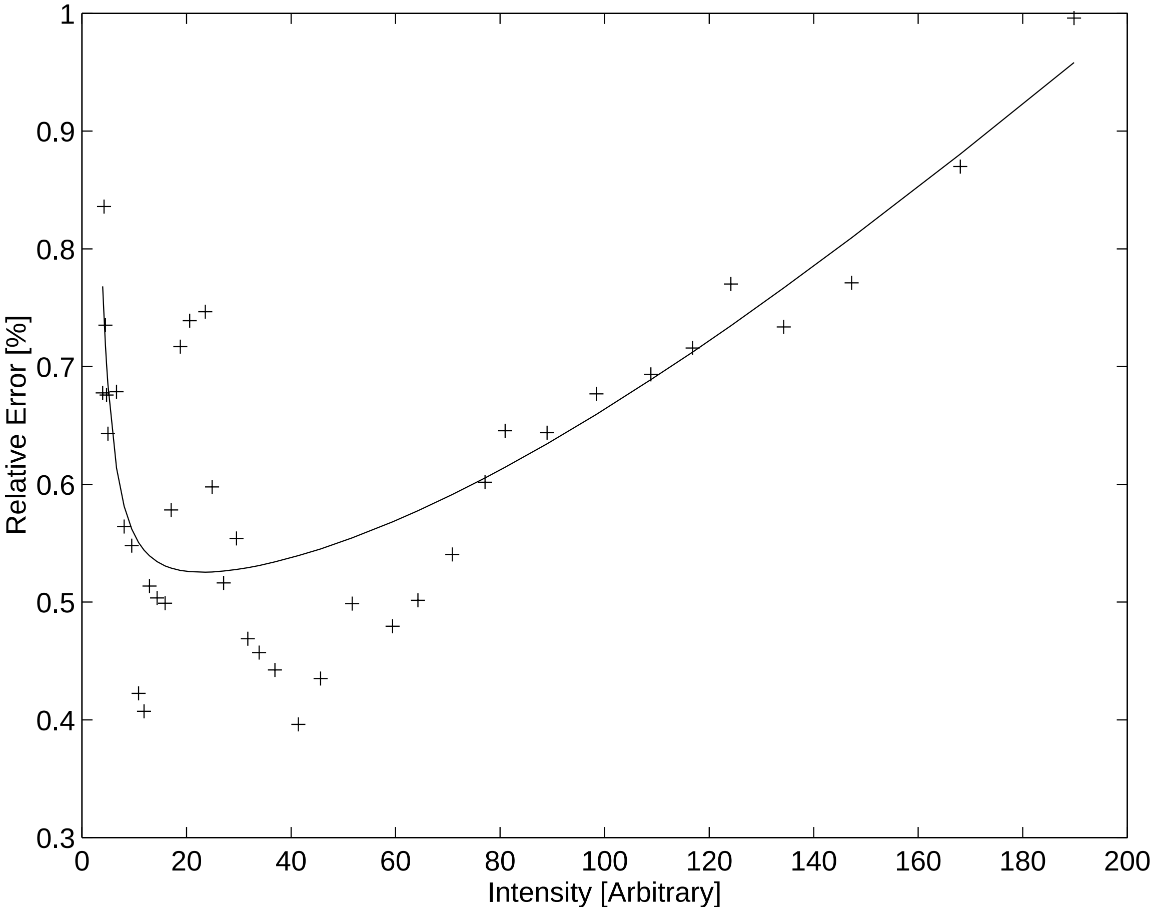


Fig. 13. Relative error σ_I/I: experimental data points and fitted error model.