A CMOS vision system on-chip with multicore sensory processing architecture for image analysis above 1,000F/s

Angel Rodríguez-Vázquez [1,2], Rafael Domínguez-Castro [1,2], Francisco Jiménez-Garrido [2] and Sergio Morillas [2]
[1] IMSE-CNM/CSIC and Universidad de Sevilla
Parque Tecnológico Isla de la Cartuja, 41092-Sevilla (SPAIN)
[2] AnaFocus (Innovaciones Microelectrónicas S.L.)
Avda Isaac Newton, Pabellón de Italia, Ático
Parque Tecnológico Isla de la Cartuja, 41092-Sevilla (SPAIN)
angel@imse-cnm.csic.es
angel.rodriguez-vazquez@anafocus.com

ABSTRACT

This paper describes a Vision-System-on-Chip (VSoC) capable of doing: image acquisition, image processing through on-chip embedded structures, and generation of pertinent reaction commands at thousand's frame-per-second rate. The chip employs a distributed processing architecture with a pre-processing stage consisting of an array of programmable sensory-processing cells, and a post-processing stage consisting of a digital microprocessor. The pre-processing stage operates as a retina-like sensor front-end. It performs parallel processing of the images captured by the sensors which are embedded together with the processors. This early processing serves to extract image features relevant to the intended tasks. The front-end incorporates also smart read-out structures which are conceived to transmit only these relevant features, thus precluding full gray-scale frames to be coded and transmitted. The chip is capable to close action-reaction loops based on the analysis of visual flow at rates above 1,000F/s with power budget below 1W peak. Also, the incorporation of processors close to the sensors enables signal-dependent, local adaptation of the sensor gains and hence high-dynamic range signal acquisition.

1 INTRODUCTION

Vision systems are among the most challenging of the application drives mentioned by ENIAC´s SRA [1] and the International Technology Roadmap for Semiconductors (ITRS) [2]. The design of imaging systems (sensors + readout + data conversion + controller + drivers) on CMOS chips has been making good progress during the last decade [3]. The main design target for CMOS imaging chips is reproducing images with given accuracy and speed. The target for vision systems is different. Similar to imagers, they have 2-D light intensity maps as inputs. Also, they may output images for monitoring purposes. However, their primary outputs are not images, but reaction commands. For instance, these commands may be needed to discard defective parts following visual inspection in a production line; or to trigger evasive maneuver following the visual detection of looming objects moving into collision course towards a vehicle; or to align unmanned aerial vehicles while landing in a platform following the signaling provided by a set of light beacons; or to trigger alert mechanisms if suspicious events are detected into a scene subjected to video surveillance; just to mention some examples.

Vision applications require to complete the full "sense → process → analyze → make decision" cycle. It involves *large amount of data*, especially in applications where high frame-rate is essential. Making a real-time decision also requires *low latency* from the system, which makes the analysis of the large input data set even more demanding.

The industrial state-of-the-art considers vision systems as "seeing computers" or "computers that see". This vision (now in the metaphoric meaning of the word) is reflected on the architecture typically used for them, namely: an *imager* (image sensor) to acquire and digitize the sensory data and a *host processor* to handle this huge amount of raw data. Such brute-force approach does completely ignore the specifics of the data, the ways how interesting pieces of information

Sensors, Cameras, and Systems for Industrial/Scientific Applications XI, edited by Erik Bodegom,
Valérie Nguyen, Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 7536, 75360O · © 2010
SPIE-IS&T · CCC code: 0277-786X/10/$18 · doi: 10.1117/12.839183

emerge from the data, and hence results in largely inefficient systems. Consider for instance the application of finding defective parts in a production line where the parts may be placed with different orientations, corresponding to up to 360º rotations. Current vision technologies can hardly go above 10F/s (Frames per second), even by using a low-resolution front-end sensor with only 128x128 pixels [4].

Not only conventional computer architectures are inadequate. Conventional algorithmic solutions used in these architectures are also inadequate. This fact has been highlighted in a very recent paper published in *Vision System Design* [5]. It states that brute force pattern matching, the conventional approach adopted by many system developers, is not the right tool in many applications. Instead, sic, "*a majority of smart camera applications can be solved using only a small number of image processing algorithms that can be learned quickly and used very effectively*" [5]. Interestingly enough these simple algorithms (thresholds, blob analysis, edge detection, average intensity, binary operators, …) can be mapped down onto dedicated, processor architectures composed of simple processors with mostly local interactions – the sort of architectures addressed by this paper.

Unconventional architectures and implementations for smart imaging chips (imagers with some embedded intelligence) and vision-dedicated chips have been reported elsewhere. For example AER silicon retina chips [6], optical flow sensors [7], visual depth sensors [8], etc. These devices include many remarkable architectural concepts and optimized circuitry and are very efficient in some *specific early-vision* tasks. Also, during the last few years the authors have relied on the concept of *visual cellular microprocessors* [9] [10] and have devised different programmable *general-purpose* early-vision chips based on this concept [11][12][13]. However, all these chips are not *autonomous* systems; i.e. they must combine with off-chip controllers and processors for completing *medium*- and *high-level* vision tasks. This paper reports a complete, autonomous Vision-System-on-Chip (VSoC) called Eye-RIS_v2.1 composed by two *multi-core* stages, which interactions are handled by an embedded controller.

2 ARCHITECTURAL CONCEPT OF THE EYE-RIS SYSTEM

Eye-RIS systems are targeted to complete vision tasks at very high-speed. For instance, to segment moving objects within a scene, compute their speeds and trajectories and provide this information to a control system which tracks the objects. This is hard task for conventional vision systems composed of a front-end sensor followed by a DSP. These architectures operate on a frame-by-frame basis. The front-end sensor must captures all frames, one by one, at the required speed; then it must read all pixel date per each frame; convert and codify all these data into a digital format; and drive a digital processor with the resulting data flow. High-speed applications require large frame rates (well above the standard video rate) and each frame is composed of a large 2-D set of data. Hence, since the digital processor must analyze a huge amount of information, either sophisticated processor architectures are employed or real-time operation becomes unfeasible.

The bottleneck of these conventional architectures is found in their frame-based operation. Reading and downloading complete frames is needed for applications whose target is reproducing full images (imaging applications), but not for vision applications. In these latter applications whole images are not important. Only those image features which convey the information required to complete the required vision tasks are. For instance, in tracking applications only the locations and speeds of the relevant objects are important. Hence, why to read-out, convert/codify and transmit full image frames? By doing so we are wasting precious resources in handling useless information (pixel data) and overloading the DSP with such useless information. In the Eye-RIS architecture this problem is overcame by incorporating processing in the sensory front-end.

The basic idea is distributing the tasks among different cores and, more specifically performing a significant part of the processing at a *front-end* section consisting of simple, tightly coupled programmable processing cores. This front-end section is realized on chip as a *multi-functional structure* with all conceptual layers implemented within a common semiconductor substrate. Relevant features of the incoming frames are extracted by this sensory-processing front-end, and only these relevant features are converted, codified and transmitted for further analysis by the DSP.

In conventional vision architectures the border between sensors and processors is placed at a point where the amount of data is large. However, in the Eye-RIS architecture this border is located at a point where the amount of data is small. Assume for illustration purposes that we target to tracking objects moving at 40m/sec into a scene. It requires capturing and analyzing images at 2000F/s rate. At the outcome of the capture/analyze process the only pertinent data is the predicted position of the objects. This is actually the only information driven to the digital processor. But to extract this information the following tasks must be completed: image acquisition; low-pass filtering; activity detection,; motion estimation; object tracking; loop control, and position prediction. In the Eye-RIS system, this is achieved by the so-called Q-Eye focal-plane processor [14].

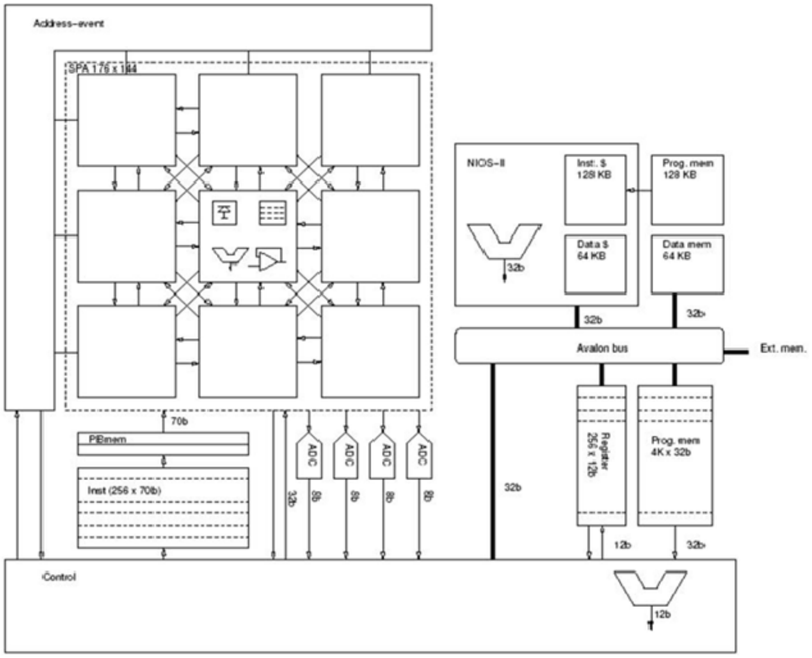

Figure 1. Eye-RIS_v2.1 chip architecture

Figure 1 shows the VSoC architecture. The pre-processor (area at the bottom left in the figure) is instrumental for performance. It computes the analog data captured by the embedded sensors and employs digital signals to control processing function, parameters and internal data flow. Digital codes are interchanged with the µP through the controller. Analog data are passed on to the µP by means of a battery of analog-to-digital converters (8bit@50MHz).

The pre-processing stage consists of an array of *multi-functional* pixels including photo-sensors, processing circuits and memories. These functions include: 2-D image sensing, image processing (a sequence of programmable space and time-domain operations are executed in the 2-D processor array); 2-D memorization of both analog and digital data; 2-D data-dependent task scheduling, control and timing, input/output operations; and storage of user-selectable instructions (programs) and parameter configurations. Pixels interchange information with their neighbors to realize a variety of operations such as:

- Linear convolutions with programmable masks.
- Time- and signal-controlled diffusions (by means of an embedded resistive grid).
- Image arithmetics.

- Signal-dependent data scheduling.
- Gray-scale to binary transformation.
- Logic operation on binary images.
- Mathematical morphology on binary images.
- Etc.

Figure 2 shows a floor-plan of the Q-Eye. The external interface is completely digital and synchronous. It is composed of a 32-bit data bus for image I/O and two additional buses, namely a 10-bits data bus and 12-bits address bus. These latter buses are employed to program a digital control system which contains 256 control words of 60-bit and individual register for analog references and miscellaneous configuration. This system controls the array of processing-sensing cells, on the one hand, and the I/O control unit which handles all basic I/O process, on the other. The I/O interface can operate in three modes:

- Loading-downloading of grey-scale images.
- Loading-downloading of binary images.
- Address-event mode.

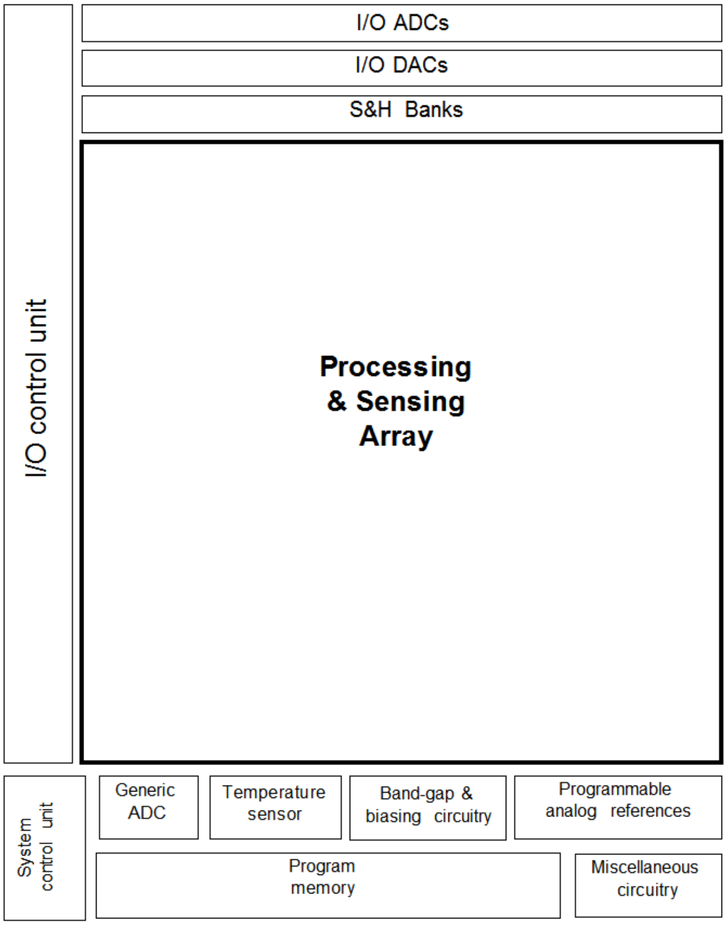

Figure 2. Florr-plan of the Q-Eye front-end.

Grey-scale values are coded into digital form by on chip 8-bits AD Flash converters, and decoded by on chip 8-bits DA resistors string converters.

To the purpose of improved *power management,* and hence reduced power consumption, most of the processing blocks in the Q-Eye and the analog reference buffers used for biasing have independent power up/down signals Also, the operation speed of most blocks is programmable. Thus, the chip can be tuned to process either very high frame rates or low frame rates with optimum power consumption for each configuration.

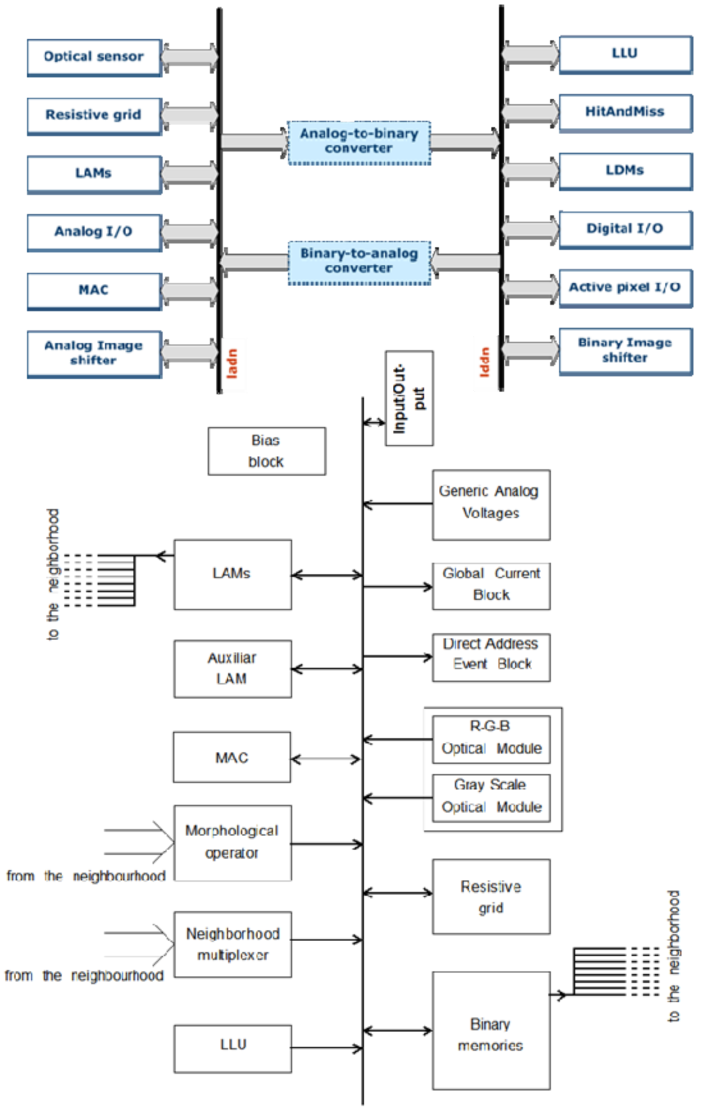

Figure 3. Two alternative conceptual representations of the block diagram of the Q-Eye sensory-processing pixel

Robustness enhancement is achieved through improved calibration techniques. Offsets in the Q-Eye are stored in static (non volatile) digital memories. Automatic calibration in the Q-Eye is performed by dedicated state machines which control in-loop A/D converters. Also, a temperature sensor and a temperature controlled correction loop are embedded in the Q-Eye to preclude the impact of junction temperature increases onto optical sensors and analog memories.

The core of the Q-Eye front end consists of an array of programmable cells. Each cell is capable to store several grey-scale and black-and-white pixel data and to perform arithmetic operations involving data of the neighbour pixels (spatial filtering) and past samples of the pixel data (temporal filtering). Figure 3 shows two alternative block diagrams for the pixel of the Q-Eye. Basic analog processing operations among pixels are *linear convolutions* with programmable masks.

The Q-Eye employs Multiplier-Accumulator Circuit unit (MAC) which processes neighbour pixels into an *algorithmic sequence*. Despite this sequential operation, computation times are similar since no calibration of the transconductors is needed.

The area saving reported by the absence of spatially replicated structures (i.e. the transconductance multipliers employed for linear convolutions [11]) enables the incorporation at the Q-Eye pixel of functions which are not found in previous chips [11]. Overall the following tasks can be realized at the front-end:

- Pixel-Wise “cosmetic operations”: Each pixel is transformed independently of its neighbors, and remains on the same location.
- Generalized convolutions: each pixel is transformed as a combination of the pixels within its neighborhood:
    - linear convolution kernels,
    - morphological operators,
    - nonlinear operations; … anisotropic diffusion, median filtering, etc.
- Movements: pixels are moved to a different position. Movements can be decomposed into shifts and rotations.
- Image-wise operations: Pixels in different images in the same or different locations are combined (either linearly or non-linearly).

3 THE EYE-RIS CHIP IN OPERATION

Besides allowing to reduce the dimensionality of data, the tight coupling between sensors and processors eases signal acquisition enhancement. Particularly, sensor gains can be controled pixel-by-pixel to allow HDR image acquisition. It is illustrated in Figure 4 where pictures at the top are acquired by suing linear integration and those at the bottom are acquired by using a well capacity adjustment algorithm [15]. DR enhancements up to 75dB are achieved. Actual enhancement can be programmed in real-time.

Figure 5 shows inputs and outcomes for different linear and nonlinear diffusion processes realized by using the *resistive grid* embedded in the Q-Eye. Both the filter type and the spatial band-width of the diffusion process can be controlled by the user. The results of performing low-pass, high-pass and band-pass spatial filtering on the input image of Figure 5(a) are shown in Figure 5(b) (c) and (d), respectively. Different values of sigma are employed. Figure 5(e) shows the output of a masked diffusion process (bottom figure), the mask being the binary figure at the top.

Figure 6 shows major steps of the processing chain implemented within the Eye-RIS_v2.1 to find defective parts in a production line. Eye-RIS_v2.1 identifies defective parts based on feature analysis instead of brute force pattern matching. It enables speed improvements (number of pieces per second) of several orders of magnitude as compared to conventional systems [4]. The figure illustrates the progressive reduction of data along the processing chain. Out of the some 26kbyte raw data acquired by the sensors at the Q-Eye front-end, only some 100byte remain after pre-processing and are actually coded in digital domain and sent to the post-processing stage. Such reduction, together with the intensive parallel processing performed at the front-end is instrumental to achieve this data reduction and hence the overall efficiency enhancement in the completion of the part finding task.

The Eye-RIS_v2.1 can be software-programmed for a large variety of applications such as distributed video surveillance networks, industrial inspection, factory automation, automotive, military, toy industry, etc. A complete set of programming tools, embedded into an application development kit [14] is available for system-level users to develop and debug specifics algorithms and programs for each application. Although programming languages are standard, applications engineers must use resources at both the pre-processing and the post-processing stages in order to take full advantage of the system capabilities.

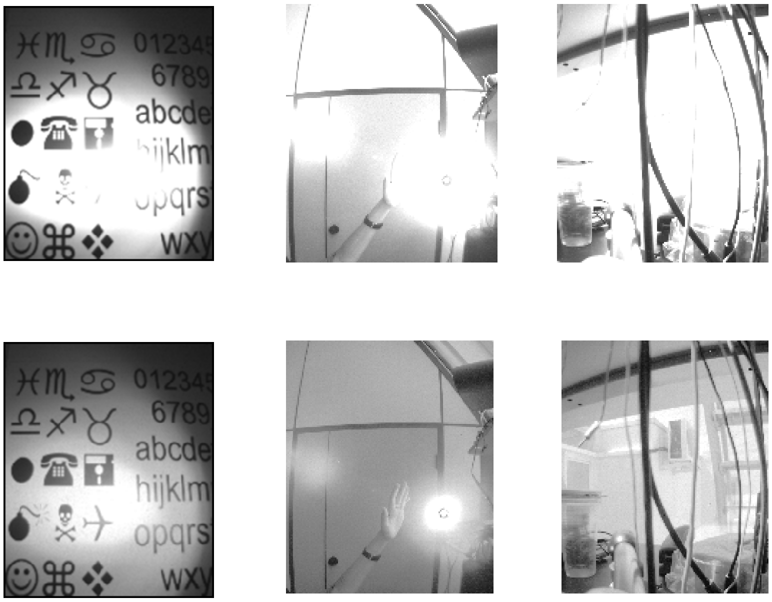

Figure 4. Acquisition of HDR images using the Eyer-RIS_v2.1. At each case, the bottom picture is acquired in linear integration mode and HDR mode, respectively. HDR acquisition is achieved by processing right at the pixel level utilizing an algorithm based on the *well capacity adjustment* technique [15].

Figure 7 illustrates a typical processing flow. It corresponds to the sequence of operations needed to guide an unmanned vehicle across a road. The splitting of tasks between pre- and post-processor illustrates the capabilities featured by the processing circuitry close to the sensors.

4 DISCUSSION

Full integration of the vision functional features into single chips represents an important stage of machine-vision system evolution. Actually, the design of imaging systems (sensors + readout + data conversion + controller + drivers) on CMOS chips has been making good progress during the last decade. In the same direction, the design of Systems-on-Chip has been also progressing during the last years. However, the design of Vision Systems on-Chip (VSoC) has not. There is hence a gap between the current art and the requirements of industry.

During the last decade, massive parallel approaches are receiving an important attention from industry and academia. Excluding dedicated chips [16], typical architectures are based on the use of multi-core processors or DSPs, where each processing core may be capable of some data-parallel operation. Devices such as Nvidia’s graphics processors (GPUs) [17], Intel’s Larrabee GPUs [18], the Cell processor [19], SPI’s stream processors [20], Tilera’s Tile64 [21], and other many-core devices [22]-[25] define, in terms of performance, current state-of-the-art in general-purpose programmable parallel computing chips. These devices are offering coarse-grain parallelism with many relatively complex autonomous or semi-autonomous processors, and are suitable for a wide range of applications requiring high computational demands.

However they are sub-optimal in terms of efficiency. For example, the high-performance Nvidia's chip consumes over 100W and features some 10 GOPs/W [17]. Alternative architectures able to extract the intrinsic parallelism of imaging applications are hence highly required.

One possible solution is resorting to *artificial retina* chips. Pioneering works on programmable artificial retina chips and focal-plane processing chips were completed already in the early 90´s [26] [27]. However, they are never reached the levels of performance and flexibility required for industrial applications. Another alternative is using digital parallel-processors with Single Instruction Multiple Data (SIMD) architecture. One dimensional (1D) SIMD processors chips have been reported elsewhere – for instance at [28] [29]. However, they do not take the full advantage of a VSoC solution since they do not embed optical sensors. Other devices including sensors [30] [31] do not address the data reduction challenge at the border between the sensing and the processing sections.

The Eye-RIS_v2.1 combines artificial retina concepts and 2D SIMD processor concepts to obtain a *general-purpose*, true VSoC with a processing capability of 250GOPS and a power consumption of 4mW/GOP. It is very well suited for applications where *compactness*, *cost,* energy *consumption efficiency* and *operation speed* define major targets.

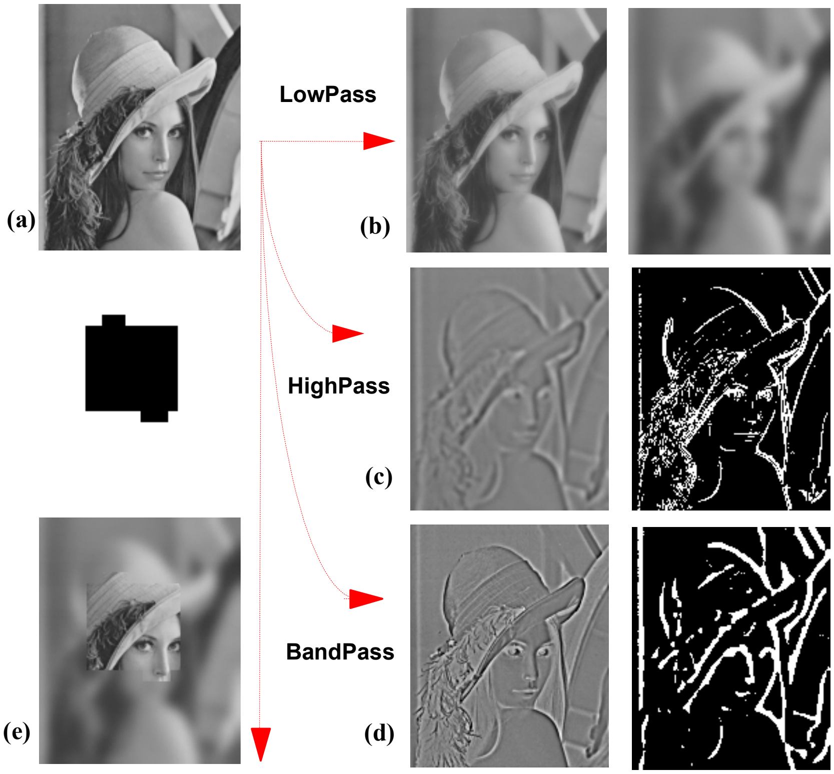

Figure 5. Illustrating the realization of different kind of spatial filterings at the Eye-RIS front-end. Lowpass: σ=1(left picture); σ=4(right picture). Highpass: σ=1.4 (left picture); BW-hp (right picture). Bandpass: σ=1.4/2.4 (left picture); BW-bp (right picture). Masked diffusion: σ=4.

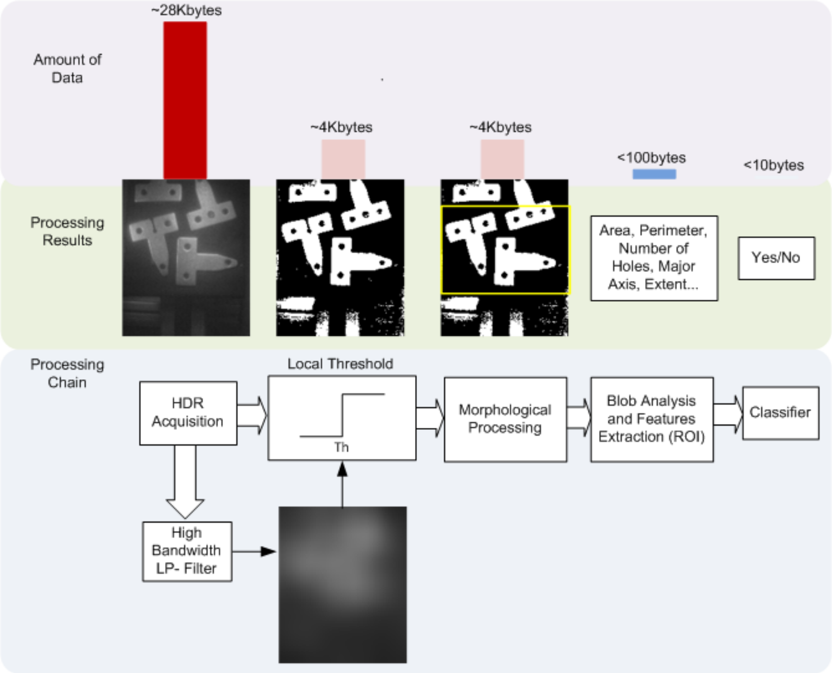

Figure 6. Illustration of the progressive data reduction of data along thevision processing chain as it actually happens in the Eye-RIS_v2.1. All steps of the processing chain above but the last one are completed in the Q-Eye sensory-processing front-end. Thus, the data set delivered for processing by the host digital processor is quite small.

5 ACKNOWLEDGEMENTS

The work of Prof. Rodríguez-Vázquez has been partially supported by the Spanish project 2006-TIC-2352 and the PIMA program of the CICE/JA. Authors would like to acknowledge fruitful discussions with Dr. Ricardo Carmona, Dr. Gustavo Liñán, Dr. Akos Zarandy and Dr. Piotr Dudek.

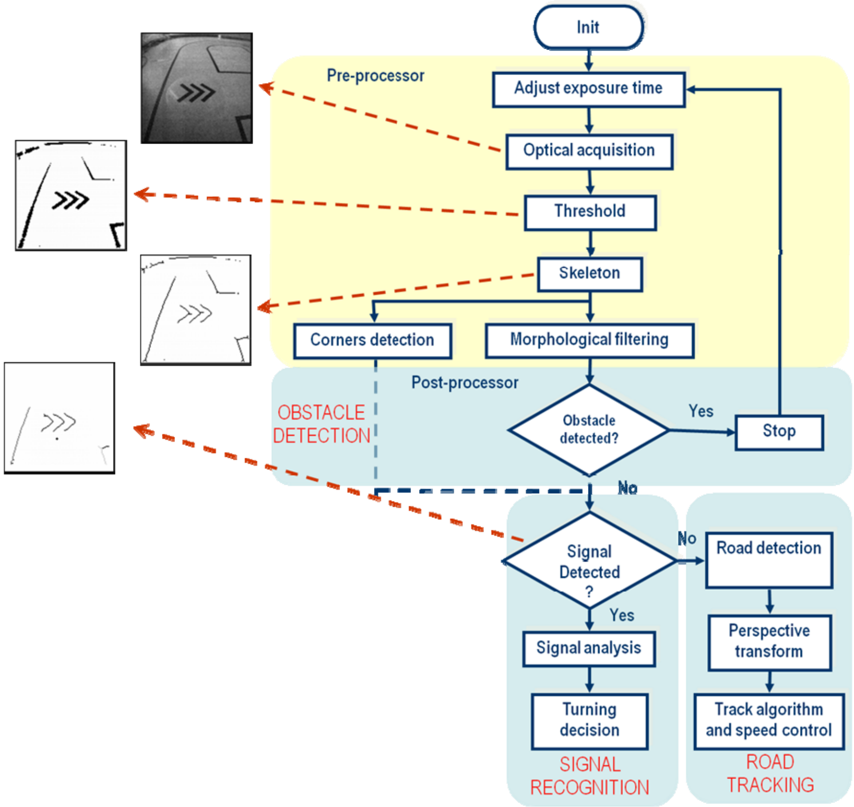

Figure 7. Typical chip processing flow

6 REFERENCES

[1] ENIAC working group, *Strategic Research Agenda* (2nd edition), European Technology Platform Initiative (2007).

[2] 2007 International Technology Roadmap for Semiconductors (ITRS) 2007 Edition Emerging Research Devices. http://www.itrs.net/Links/2007ITRS/Home2007.htm.

[3] A. El Gamal and H. Eltoukhy, "CMOS Image Sensors," IEEE Circuits and Devices Magazine, 6-20 (2005).

[4] Cognex Ltd, http://www.cognex.com/ProductsServices/InspectionSensors.

[5] G. Devaraj et al., "Applying Algorithms," Vision System Design 13(11), 17-20 and 85-87(2008).

[6] T. Delbruck and P. Lichsteiner, "Freeing Vision from Frames," The Neuromorphic Engineer, 3, 3-4 (2006).

[7] W.E. Green et al., "Flying Insect-Inspired Vision for Autonomous Aerial Robot Maneuvers in Near-Earth Environment," Proc. IEEE Intl Conf. on Robotics and Automation, 2347-2352 (2004).

[8] R. M. Philipp, et al., "A 128×128 33mW 30frames/s Single-Chip Stereo Imager," IEEE Int. Solid-State Circuits Conference, 2050-2059 (2006).

[9] L. O. Chua and T. Roska, [Cellular Neural Networks and Visual Computing], Cambridge University Press, Cambridge UK (2002).

[10] T. Roska and A. Rodríguez-Vázquez, [*Towards the Visual Microprocessor*], John Wiley and Sons, Chichecter UK (2001).

[11] A. Rodríguez-Vázquez et al., "ACE16k: The Third Generation of Mixed-Signal SIMD-CNN ACE Chips Toward VSoCs," IEEE Transactions on Circuits and Systems-I, 51(5), 851-863 (2004).

[12] G. Liñán et al., "A 1000FPS@128x128 Vision Processor with 8-Bit Digitized I/O, " IEEE Journal of Solid-State Circuits, 39(7), 1044-1055 (2004).

[13] R. Carmona et al., "A Bio-Inspired 2-Layer Mixed-Signal Mixed-Signal Flexible Programmable Chip for Early Vision," IEEE Transactions on Neural Networks, 14(5), 1313-1336 (2003).

[14] AnaFocus Ltd, http://www.anafocus.com

[15] S. J. Decker et al., "A 256x256 CMOS Imaging Array with Wide Dynamic Range Pixels and Column-parallel Digital Output," IEEE J. of Solid State Circuits, 33(12), 2081-2091 (1998).

[16] T. Komuro et al., "A Digital Vision Chip Specialized for High-Speed Target Tracking," IEEE Transactions on Electron Devices, 50 (1), 191-199 (2003).

[17] K. Morris, "A Passel of Processors: NVIDIA's Tesla T10P Blurs Some Lines, " FPGA and Structured ASIC Journal, (2008).
(on-line: http://www.fpgajournal.com/articles_2008/20080617_nvidia.htm)

[18] L. Seiler et al., "Larrabee: A Many-Core x86 Architecture for Visual Computing," ACM Transactions on Graphics, 27 (3), (2008).

[19] D. Pham, et al. "The Design and Implementation of a First-Generation CELL Processor," IEEE International Solid-State Circuits Conference Dig. Tech. Papers, 184–185 (2005).

[20] U.J. Kapasi et al., "Programmable Stream Processors," IEEE Computer, 36 (8), 54-62 (2003).

[21] D. Wentzlaff et.al. "On Chip Interconnection Architecture of the TILE Processor," IEEE Micro, 27 (5), 15-31 (2007).

[22] S. Vangal et al. "An 80-Tile 1.28TFLOPS Network-on-Chip in 65nm CMOS," International Solid-State Circuits Conference Dig. Tech. Papers, 98–99 (2007).

[23] A.Duller et al., "Parallel Processing -the picoChip way!," Communicating Process Architectures, IOS Press, 125-138 (2003).

[24] Z.Yu et al. "Architecture and Evaluation of an Asynchronous Array of Simple Processors," Journal of Signal Processing Systems, 53 (3), 243–259 (2008).

[25] Keckler, S., et al. "A Wire-Delay Scalable Microprocessor Architecture For High Performance Systems," IEEE International Solid-State Circuits Conference Dig. of Tech. Papers, 168–169 (2003).

[26] T. Bernard et al., "A Programmable Artificial Retina," IEEE J. Solid State Circuits, 28 (7), 789–797 (1993).

[27] J. Eklund et al., "VLSI Implementation of a Focal Plane Image Processor—A Realization of the Near-Sensor Image Processing Concept," IEEE Transactions on VLSI Systems, 4 (3), 322–335 (1996).

[28] A.A. Abbo et al., "Xetal-II: A 107 GOPS, 600 mW Massively Parallel Processor for Video Scene Analysis," IEEE J. Solid-State Circuits, 43 (1), 192-201 (2008).

[29] S. Kyo et al, "An Integrated Memory Array Processor Architecture for Embedded Image Recognition Systems," Proceedings of the 32nd International Symposium on Computer Architecture (ISCA'05), 134-145 (2005).

[30] Chih-Chi Cheng et al., "iVisual: An Intelligent Visual Sensor SoC with 2790fps CMOS Image Sensor and 205GOPS/W Vision Processor," IEEE Int. Solid-State Circuits Conference Dig. Tech. Papers, 306-307, (2008).

[31] L.Lindgren et al., "A Multi-resolution 100-GOPS 4-Gpixels/s Programmable Smart Vision Sensor for Multi-sense Imaging," IEEE Journal of Solid-State Circuits, 40 (6), 1350-1359 (2004).