

BI-I: A STANDALONE ULTRA HIGH SPEED CELLULAR VISION SYSTEM

Ákos Zarándy^{1,3} and Csaba Rekeczky^{2,3}

¹Analogical and Neural Computing Laboratory
Computer and Automation Institute
Hungarian Academy of Sciences, Budapest, Hungary

²Faculty of Information Technology
Péter Pázmány Catholic University, Budapest Hungary

³AnaLogic Computers Ltd, Budapest, Hungary

ABSTRACT: The Bi-i standalone cellular vision system is introduced and discussed. In the first part, the underlying sensor and system level architectures are presented and various implementations will be overviewed. This computing platform consists of state-of-the-art sensing, cellular sensing-processing, digital signal processing and communication devices that make it possible to use the system as an ideal computing platform for combined topographic and non-topographic calculations in sensing-processing-actuation scenarios. In the second part of the paper, ultra high frame rate laboratory experiments will be shown and discussed to highlight the most peculiar features of the system and its applicability in various industrial quality control areas. The overview will also underline the potentials of the Bi-i vision system for unmanned intelligent vehicle type applications in visual exploration, identification, tracking and navigation.

1. Introduction

Real-time inspection of textile fibers, spark plugs or pills during a fast manufacturing process was an unsolved task in the recent past. Similarly, a visual control of an ultra-fast robotic assembly process, multi-object laser tracking or visual terrain identification and navigation on an autonomous intelligent vehicle all belong to the problem areas where traditional systems consisting of (off-the-shelf) CCD/CMOS cameras connected to (PC like) computers could not provide the required solutions.

All the above complex application areas envision a standalone platform with integrated sensing-processing capability efficiently combining topographic¹ and non-topographic² computing. Different versions of such architecture, called the Compact Cellular Visual Microprocessor (COMPACT CVM), have been designed (*Fig. 3*) and are currently being tested. These bio-inspired (Bi-i³) standalone vision systems largely build on the results of cellular neural network (CNN,[1]-[2]) theory, CNN chip prototypes (e.g. [7]-[9]), retinal analysis and modeling (e.g. [3]-[6]), and various bio-inspired topographic schemes developed for image flow processing (e.g. [10]-[12]).

¹ In this context: 2D sensor-structure related array computation

² In this context: vector-matrix computation

³ Bi-i is the registered trademark of AnaLogic Computers Inc.

In this overview paper after reviewing the CNN basics and the CNN-UM visual microprocessor architecture, we will describe the Bi-i standalone ultra high speed vision system along with the special experiments and application areas that could benefit from using this standalone multiple sensor-processor-actuator system.

2. CNN Basics and the Associated Sensor-processor Architectures

Coupled nonlinear ODEs define a cellular neural/nonlinear network (CNN, [1]-[5], see also a computing methodology driven introduction to CNNs in the companion paper [17]) structure composed of simple dynamical systems, called cells arranged at the node points of a regular grid. The state of any cells is represented by a real number, their interaction is local and the internal operation is continuous in time.

The cell dynamics is described by the following nonlinear ordinary differential equation with inter-cell coupling (the extension to higher dimensions is straightforward, allowing similar interlayer interactions):

$$C \frac{d}{dt} x_{ij}(t) = -R^{-1} x_{ij}(t) + \sum_{kl \in N_r} A_{ij,kl} y_{kl}(t) + \sum_{kl \in N_r} B_{ij,kl} u_{kl}(t) + z_{ij}$$

$$y_{ij}(t) = f(x_{ij}(t))$$

where x_{ij} , u_{ij} , y_{ij} are the state, input and output voltage of the specified CNN cell, respectively. The notation ij refers to a grid point associated with a cell on the 2D $M \times N$ grid, and $kl \in N_r$ is a grid point in the neighborhood within the radius r of the cell ij . $A_{ij,kl}$ and $B_{ij,kl}$ represent the linear feedback and the linear control, respectively. The constant z_{ij} is the cell current, which could also be interpreted as a space-varying threshold. In general, the CNN template, which is the “program” of the CNN array, consists of the $[A \ B \ z]$ terms (omitting the indices). The output characteristic f is a sigmoid-type (e.g. piecewise-linear) function. The time constant of a (first order) CNN cell is determined by the linear capacitor (C) and the linear resistor (R) and it can be expressed as $\tau = RC$. Without loss of generality $R = 1$ and $C = 1$ can be considered.

Remarks:

- (i) The above equation describes the standard CNN model ([1]). As it will be discussed later both the cell and the interaction types have been further generalized for numerous application areas.
- (ii) Mixed-signal CMOS physical implementations with gray-scale I/O closely follow this dynamics, except for some modifications limiting the dynamic range of the cell’s state value (“full-range” model, [8], [9])

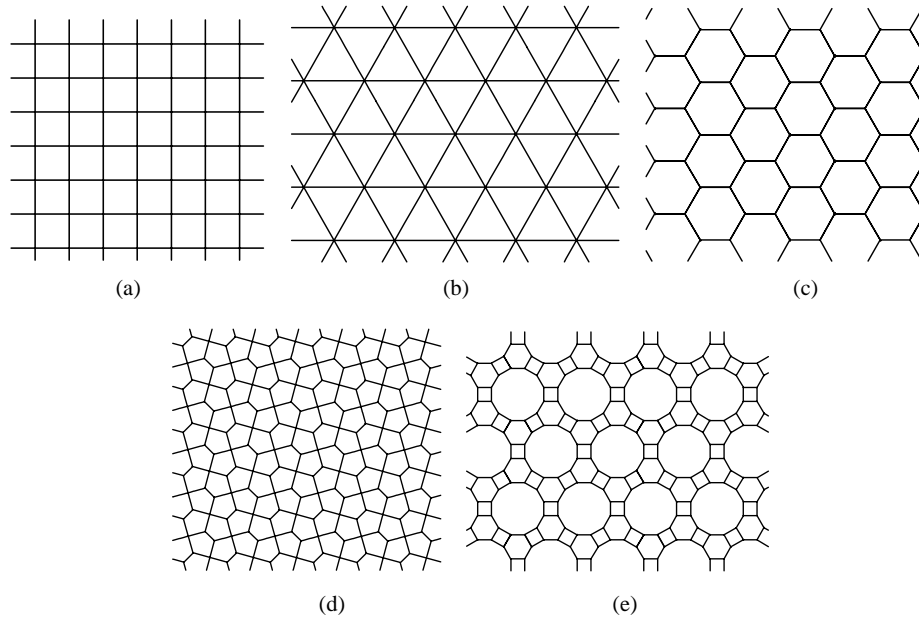


Fig. 1 The most common and two special grid-types of cellular architectures. Regular contiguous tessellations of the plain based on congruent polygons are shown in the top row: (a) rectangular, (b) triangular and (c) hexagonal lattices. In the bottom row two special grid-types are given that belong to the ornamental group. All grid-types (b)-(e) could be mapped to a rectangular grid (a) with periodic space-variant interconnections.

The preceding definition of CNNs can be further generalized and several types of specific cellular (nonlinear) arrays can be generated including a 2D cellular automata (CA, [15]).

They can be classified according to the types of the *grid*, the *processor* (cell), the *interaction* (template or update rule), and the *mode of operation*. Thus, building the core of a massively parallel computer based on cellular array architecture; the following key points should be addressed in detail:

Grid types:

Cellular arrays are usually defined on a spatially discrete *square* (rectangular) grid; however, *hexagonal* and *triangular* arrangements can also be considered (Fig. 1 (a)-(c)). These grids are the only regular contiguous tessellations of the plain based on congruent polygons alone. Other grid-types can also be created based on non-regular congruent polygons or from a regular vertex grid through discrete geometrical transformations: rotations and translations (ornamental groups, Fig. 1(d)-(e)). A great number of these grids can be mapped on a typical eight-neighbor rectangular structure with periodic space-variant connections [14]. Multiple and varying grid sizes (e.g. course and fine grid; logarithmically changing size, etc.) may be useful in simulating adaptive biological systems (e.g. retina [4], LGN or magno/parvo pathways in the cortex).

Cell (processor) types and boundary cell types:

Cellular arrays can be built from linear/nonlinear, first or higher order cells. A linear or “small-signal” operation (as a special case) is achievable through a piece-wise linear output characteristic. A highly nonlinear dynamic behavior (typical CNN models) is attainable through a sigmoid; Gaussian or inverse-Gaussian type output characteristics. Cells can

include additional local analog and/or logical memories for storing intermediate processing results. A cell might perform local logic operations (typical CA models) or a combined analogic (analog and logic) computation (extended CNN models). The cells of an array can be uniform or non-uniform (regularly or slightly varying). In some processing tasks two or three cell types in a regular grid might be very useful, for example, in color image processing.

Since in any physical implementation only a finite array can be built, a boundary condition should be exactly defined. Creating boundary cells that are not involved in the cooperative computation (“virtual cells”) can satisfy this requirement. The most important boundary cell specifications are as follows: (i) *Fixed (Dirichlet)*: constant values are assigned to all boundary cells; (ii) *Zero-flux (Neumann)*: boundary cells are made to follow cells that are on the same side of the array; and (iii) *Periodic (Toroidal)*: boundary cells are made to track the values of cells from the opposite side of the array.

Neighborhood-size and interaction types:

The interaction type (the biological “synapse”) in between the grid cells represents the program of a cellular array. Depending on whether these interaction types are fixed or programmable, the array can be regarded as a specific purpose or a re-configurable parallel array architecture.

The nearest neighborhood is the most common sphere of influence in the inter-cell communication for both CA and CNN models (either the cross-shaped 4-connected or the star-shaped 8 connected neighborhoods), however, larger neighborhood sizes can also be considered (typically in biological modeling or when creating adaptive artificial systems).

The interaction types can be linear or nonlinear memoryless functions (e.g. a typical CA update rule) of one, two or more variables. Delayed-nonlinear (e.g. a typical CNN template) and dynamic (lumped) interactions are more general connection types. Breaking the symmetry or isotropy, furthermore, varying the nature of the interconnections in space and/or in time an extremely rich collection of further templates (or local update rules) can be created.

Modes of the operation:

The modes of operation can be continuous (general CNN models) or discrete time (general DTCNN and CA models). In the case of a discrete-time procedure the update mechanism can be synchronous or asynchronous. The computation can be analog, logic or analogic, and can be executed either in local mode or in propagating mode (on a de-coupled or coupled array, respectively). Besides the usual fixed-point operational mode (equilibrium computing), transient (non-equilibrium computing), oscillating, chaotic and general stochastic modes can also be exploited.

CNN-UM architecture:

All early neural network chip realizations had a common problem: they implemented a single instruction only, thus the weight matrix was fixed when processing some input. Reprogramming (i.e. changing the weight matrix) was possible for some devices but took in order of magnitudes longer time than the computation itself.

This observation motivated the design of the CNN Universal Machine (CNN-UM, [2]), a stored program nonlinear array computer. This new architecture is able to combine analog array operations with local logic efficiently⁴. Since the reprogramming time is approximately equal to the settling time of a non-propagating analog operation it is capable of executing

⁴ In a broader sense the CNN-UM architecture can also be viewed as a general prototype architecture for both analog and digital stored program cellular distributed computing.

complex analogic algorithms. To ensure programmability, a global programming unit was added to the array, and to make it possible an efficient reuse of intermediate results, each computing cell was extended by local memories. In addition to local storage, every cell might be equipped with local sensors and additional circuitry to perform cell-wise analog and logical operations. The architecture of the CNN-UM is shown in Fig. 2.

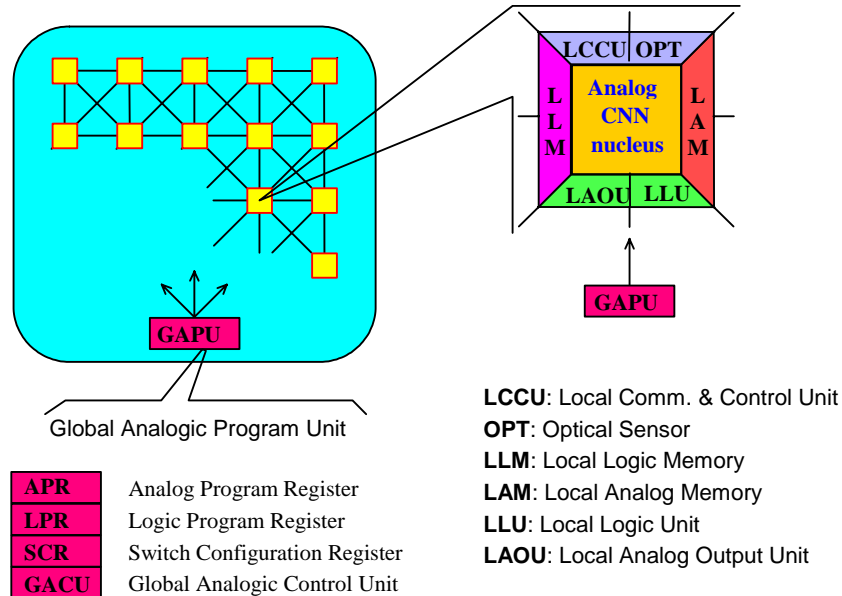


Fig. 2 The architecture of the CNN Universal Machine, the analogic array supercomputer. The figure shows the elements in the complex CNN Nucleus and the functional blocks of the Global Analogic Programming Unit.

As illustrated in Fig. 2 the CNN-UM is built around the dynamic computing core of a simple CNN. An image can be acquired through the sensory input (e.g. OPT: Optical Sensor). Local memories store analog (LAM: Local Analog Memory) and logic (LLM: Local Logical Memory) values in each cell. A Local Analog Output Unit (LAOU) and a Local Logic Unit (LLU) perform cell-wise analog and logic operations on the stored values. The output is always transferred to one of the local memories. The Local Communication and Control Unit (LCCU) provides for communication between the extended cell and the central programming unit of the machine, the Global Analogic Programming Unit (GAPU). The GAPU has four functional blocks. The Analog Program Register (APR) stores the analog program instructions, the CNN templates. In case of linear templates, for a connectivity $r = 1$ a set of 19 real numbers have to be stored (this is even less for both linear and nonlinear templates assuming spatial symmetry and isotropy). All other units within the GAPU are logic registers containing the control codes for operating the cell array. The Local Program Register (LPR) contains control sequences for the individual cell's LLU, the Switch Configuration Register (SCR) stores the codes to initiate the different switch configurations when accessing the different functional units (e.g. whether to run a linear or nonlinear template). The Global Analogic Control Unit (GACU) stores the instruction sequence of the main (analogic) program. The GACU also controls the timing, sequence of instructions and data transfers on the chip and synchronizes the communication with any external controlling device.

Synthesizing an analogic algorithm running on the CNN-UM the designer should decompose the solution in a sequence of analog and logical operations. A limited number of intermediate results can be locally stored and combined. Some of these outputs can be used as

a bias map (space variant current) or fixed-state map (space-variant mask) in the next operation adding spatial adaptivity to the algorithms without introducing complicated inter-cell couplings. Analog operations are defined by either a linear or a nonlinear template. The output can be defined both in fixed and non-fixed state of the network (equilibrium and non-equilibrium computing) depending on the control of the transient length. It can be assumed that elementary logical (Not, And, Or, etc.) and arithmetical (Add, Sub) operations are implemented and can be used on the cell level between LLM and LAM locations, respectively. In addition data transfer and conversion can be performed between LAMs and LLMs.

There is of course always a gap in between the system level design and chip implementations. Trying to synthesize powerful algorithms that can keep up with the state-of-the-art methods of signal and image processing disciplines the engineer at the system level always faces the problem of hardware limitations and have to simplify the methodologies used. On the other hand, a VLSI designer would like to encounter the priority of requirements motivated by different applications. Some of these compromises were successfully met in the ACEx CNN-UM chip design roadmap ([16], see also [7]-[9]).

In the sequel we will describe the COMPACT CVM architecture based Bi-i vision system that incorporates the most complex fully functional CNN-UM type microprocessor built up-to-date (ACE16k [9], [16]).

2. Bio-inspired (Bi-i) Vision System Architecture

In the sequel the COMPACT CVM will be introduced along with all major system components and functional block diagrams. The bio-inspired (Bi-i) COMPACT CVM architecture (*Fig. 3*) builds on state-of-the-art CNN type (ACE16k) and DSP type (TX 6x) microprocessors and its algorithmic framework contains several feedback and automatic control mechanisms in between different processing stages (*Fig. 4*). The architecture is standalone and with the interfacing communication processor it is capable of a 100 Mbit/sec information exchange with the environment (over TCP/IP). The COMPACT CVM is also reconfigurable, i.e. it can be used as a monocular or a binocular device with a proper selection of a high resolution CMOS sensor (IBIS 5x) and a low resolution CNN sensor-processor (ACE16k). The high resolution sensor could be used as a random access device (with built in 1:2 digital zoom) acquiring CNN chip size images from algorithm controlled positions at different time instants. With these two sensors, having the same field of view, one can also design algorithms that make use of three integer numbered spatial scales (roughly 1:2:8 at 128x128 spatial resolution) while navigating in a 1280x1024 high resolution projection of the surrounding world at a temporal rate of up to a few thousand frames/sec.

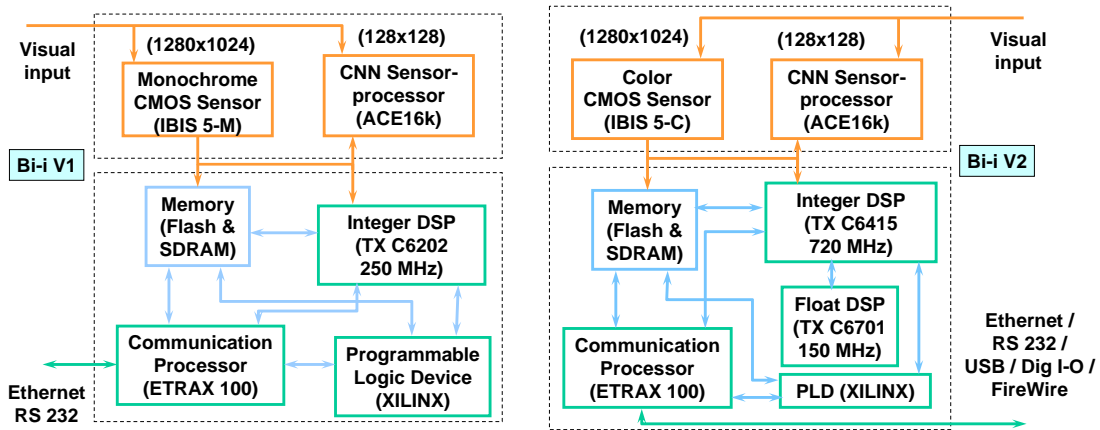


Fig. 3 The main hardware building blocks of the Bio-inspired (Bi-i) COMPACT CVM architecture. The figure shows two different versions of this standalone system both consisting of a sensor platform and a control platform. The first prototype (in the left: Bi-i V1) contains a high-resolution monochrome CMOS sensor array (IBIS 5-M), a low-resolution cellular sensor-processor (ACE16k), an integer digital signal processor (TX C6202) and a communication processor (ETRAX 100). In contrast the second prototype (in the right: Bi-i V2) has a color CMOS sensor array (IBIS 5-C) and two high-end digital signal processors (TX C6415 and TX C6701). While both systems run an embedded Linux on the communication processor Bi-i V2 has more complex external interfaces (USB, FireWire and a general digital I/O in addition to the Ethernet and RS232 built also into the Bi-i V1).

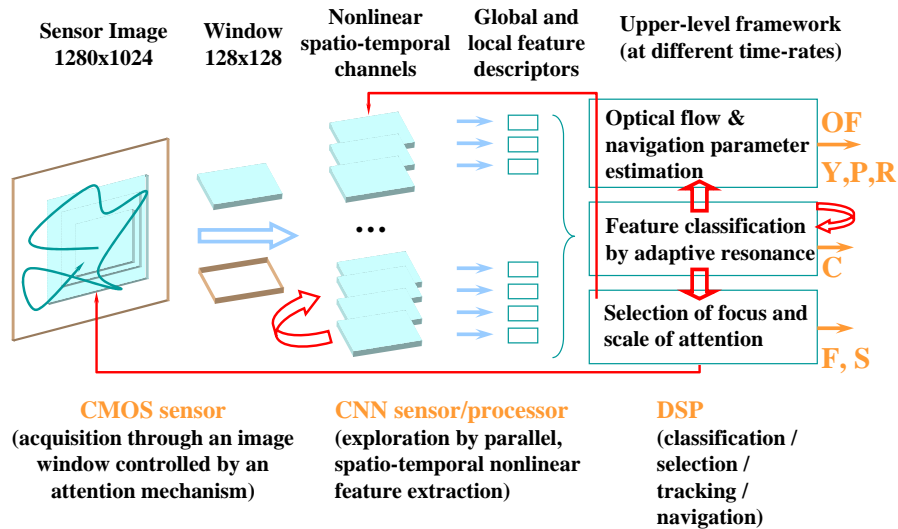


Fig. 4 General flow-processing diagram of the COMPACT CVM algorithmic framework. The image frame acquired by the CMOS sensor array is never evaluated at full resolution, only a few sub-frames are processed identical to the resolution of the CNN sensor-processor (the figure shows a single sub-frame acquisition in the active fovea of the system controlled by an upper level attention mechanism). In the first step of the evaluation the selected image window is processed by topographic parallel nonlinear spatio-temporal algorithms. Then local and global features are extracted and forwarded to the upper level where non-topographic calculations take place at different time rates. The peculiar property of the entire scheme is the existence of several feedback mechanisms in between different levels of processing that makes it possible that top-down and bottom-up processing is combined within this framework.

In summary, the COMPACT CVM architecture can be attributed as follows (the specifications are related to the associated Bi-i V2 systems):

- **High speed, compact and (relatively) low-power system**

The entire system consumes around 5-15 Watts (component dependent, the peak value is mainly due to the high power consumption of the ACE16k chip in the range of 3-6 Watts depending on the operating mode), however it delivers an overall computing power (approximately 1 GFLOPS / 6000 MIPS digital and 1 TOPS analog) that is significantly higher than the performance of any competing standalone vision systems. In the computer complexity space (Speed-Power-Volume measure) it can be stated that the system is relatively “low-power” (the volume of the system with packaging and optics is less than 1000 cm³ and weighs less than 1 kg).

- **Fault-tolerant visual computer**

The architecture is “fault-tolerant” at the topographic cellular processing level, i.e. a low number of cells not functioning properly will not significantly affect the overall performance. This has been tested by masking a certain number of cells during the operation and also by using chips for which the yield has not been 100% for all cells.

- **Multi-task computing platform**

- o Multiple attention and selection mechanisms
- o Multi-channel nonlinear feature detection based terrain classification
- o Multi-grid motion analysis based navigation parameter estimation
- o Multiple target detection and tracking

- **Biologically inspired sensor-computer architecture**

The system provides hardware support for both topographic and non-topographic computing at different levels of analysis. This platform makes it possible that a broad class of bio-inspired processing strategies can be developed and tested within a unified framework.

In this section we have described the COMPACT CVM architecture and have argued that it is a fault tolerant multi-task visual computer and an ideal computational platform for high speed bio-inspired exploration, classification, selection, tracking and navigation (see further details in [20]). In the sequel we will focus on ultra high frame rate applications to highlight the most peculiar capabilities of the Bi-i system.

3. Ultra High Speed Processing Examples

The unique feature of the Bi-i system resides in its ability of capturing and processing over 10,000 frames of 128x128 sized images in a second. The key component to reaching this exceptional performance is the focal plane CNN-UM type array processor chip (ACE16k [9]). Depending on the complexity of the problem to be solved this overall sensing-processing speed can significantly vary and at extreme cases (such as image acquisition in a strongly illuminated environment and a subsequent simple shape classification) could reach and even go beyond to 30,000 fps (1000 times the standard video frame rate!).

In this section we describe four different ultra high speed application examples of the Bi-i system and provide the related measurement results. In the first three examples, the high frame-rate sensing-processing capability of the ACE16k chip is exploited. In the last example, the high resolution CMOS sensor is used for image acquisition and the ACE16k chip is employed as a co-processor during the analysis.

3.1 Small Object Classification⁵

This example deals with the classification of small objects (like pills, grains, nuts, nails, etc.) moving at very high speeds. The capability of the Bi-i system is demonstrated using a pill detection and classification example.

Experimental setup: Objects (pills and capsules) of varying sizes and shapes are placed along the perimeter of a rotating disc. The Bi-i is positioned with its sensor-processor array focusing to a point in the path of the pills (with only one pill in its view) while the disc is rotated at a speed of over 15 m/s (see Fig. 5). In the current setup the images can be captured with 20 microsecond exposure time. The task is to classify each object based on size and shape, and to detect if the deviation from the expected shape is larger than a certain threshold (which indicates that a piece of the pill/capsule is missing).

Algorithmic solution: Instead of using electrical synchronization between the disk and the Bi-i system, a different strategy is applied based on heavy over-sampling. In the algorithm, first a quick decision is made, whether a new object is fully within the analysis window. If this is the case, the captured gray-scale image is converted to black-and-white and the size, shape and position parameters are calculated. The classification is based on the size and shape parameters of different pills (Fig. 5). Then, a circle is generated and compared to the acquired object, and possible missing parts are identified. Among these steps, only the model circle generation and the final classification are performed by the DSP; the remaining computational steps are done in parallel on the ACE16k chip.

Results: The current implementation of the small object classification algorithm runs on Bi-i V1. Overall, the system processes over 2,000 objects in a second and the sensing-processing-classification for 9 objects runs above 25,000 fps. The program can be easily adapted to classification and quality inspection of other kinds of objects (e.g. foreign particles in some liquid, living cells of various shapes in some population, etc.).

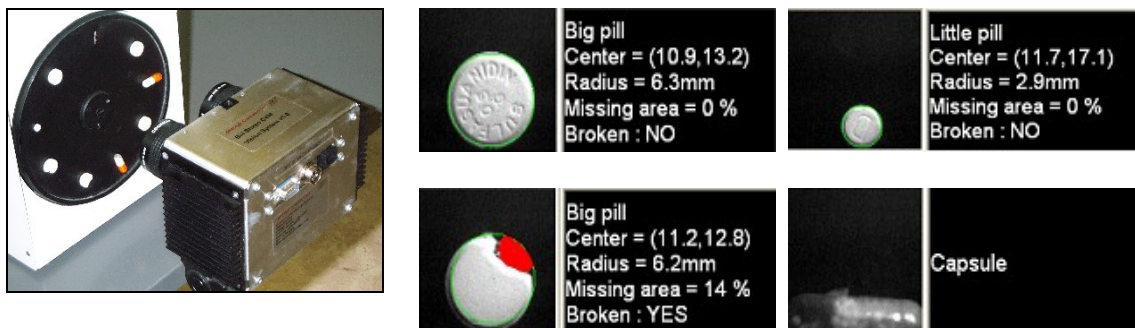


Fig. 5 Ultra-high frame rate pills analysis with the Bi-i system. Experimental setup with the Bi-i V1 system (in the left) and the display screen showing some final analysis results (in the right).

⁵ The first version of this experiment is described in details in [13] using a previous generation CNN-type microprocessor ([7]).

3.2 Spark Property Characterization⁶

This example focuses to the analysis of sparks generated at very high frame rates. Spark analysis could be important in the inspection of spark plugs, and also in the testing of high voltage ceramic insulators. Real-time spark analysis, beyond doubt, requires an ultra high frame-rate, because the length of a spark is usually below one millisecond and in order to capture the spatio-temporal event of generation a significant over-sampling is needed. In the example to be briefly described here the capability of the Bi-i system is shown in a laboratory spark-plug analysis example (*Fig. 6*). The task is to capture sparks produced by a spark plug and characterize them according to various criteria in real-time.

Experimental setup: In the experimental set up, the sensor-processor of the Bi-i system points to the area of interest in the spark plug while it is operating (see *Fig. 6*). Sparks can exhibit different characteristics, which can be indications of certain types of faults. Examples of typical sparks are shown in the pictures below.

Algorithmic solutions: The algorithmic solutions developed are based on fast binary morphology based processing executed by the Ace16k chip and make it possible to perform the following measurements on the images captured:

- spark event detection
- calculation of an aggregate image of a spark train
- morphological shape analysis
- identification of split sparks
- position monitoring of sparks
- statistical analyses of the locations, splitting and duration
- energy estimation of sparks

Results: The current implementation of the spark plug inspection algorithm runs on Bi-i V1. The local sensor memory of the ACE16k chip allows the storage of 8 consecutive images, which helps to increase the image capturing speed up to 200,000 frames per second (for up to 8 frames). The images can then be transferred off-chip for further analysis. In this procedure the Bi-i system is capable of classifying a spark at a rate of over 50,000 frames per second. The focal plane array enables the system to perform morphological operations without significant speed reduction. This high-performance event detection capability with subsequent acquisition and analysis could also be exploited in the analysis of similar extremely fast events of short duration.

⁶ The first version of this experiment is described in details in [13] using a previous generation CNN-type microprocessor ([7]).

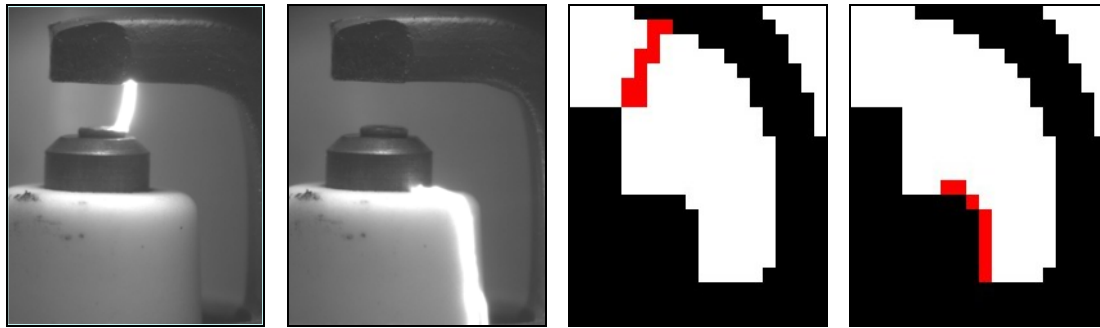


Fig. 6 Ultra-high frame rate spark plug analysis with the Bi-i system. Gas and surface discharges should be distinguished. The images in the left show typical discharges (high resolution gray-scale), the images in the right the detected and classified events (low resolution binary).

3.3 Non-contact Speed and Displacement Measurement of Surfaces

Visual speed and displacement measurement of surfaces is important in many areas, where the material (textile, paper, iron, etc) is very thin or slippery. In these cases using wheeled displacement measurement devices could cause damage and/or would provide inaccurate results. In the following example the capability of the Bi-i system is demonstrated performing non-contact speed and displacement measurements of textile surfaces.

Experimental setup and the algorithmic solution: In the experimental set up, the sensor-processor of the Bi-i system points to the area of interest and captures the textile patterns. The basic algorithmic idea is to capture two snapshots with a known time difference, and measure the displacement in pixels. Thus, an area is selected on the “entering” side of the captured image (Fig. 7), and the location of the selected area in the subsequent snapshots is identified when reaching the other end of the image. Then, the displacement and the associated time difference is derived, and the corresponding textile speed is calculated. This solution also requires a heavy over-sampling as in the previous examples, especially, in case of a periodic material, like a dense textile with homogeneous patterns.

Results: The current implementation of the speed measurement algorithm runs on Bi-i V2 and the maximum measurable linear speed is determined as 5 m/sec.

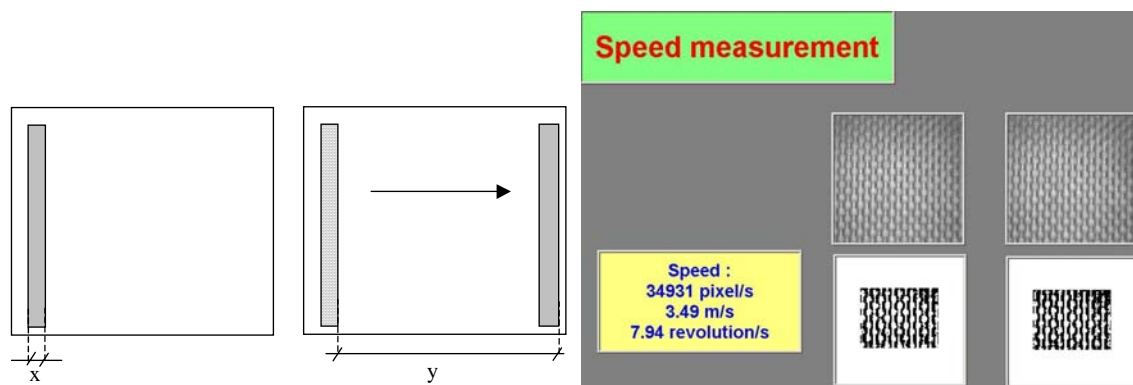


Fig. 7 Texture speed measurement with some typical examples. The figures in the left show the processing principle, the figure in the right displays a snapshot with the acquired and processed binary images.

3.4 Non-contact Texture Fault Detection

The Bi-i system can also be used in industrial quality control applications as part of a larger system producing narrow fabric items such as ribbons and labels. During the manufacturing process up to 15 cm long labels by 8 cm wide strips are woven. Later these textile labels are typically sewn onto different kind of clothes. A possible goal of the experiments is the development of a visual inspection system that identifies different kind of faults on the labels and sends messages to the control system of the machine. This prototype is described in the sequel.

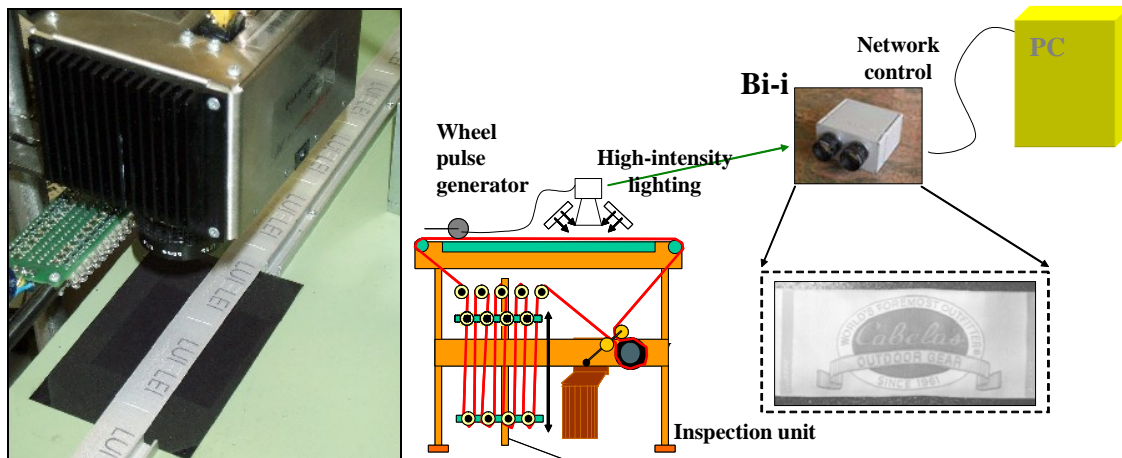


Fig. 8 Texture fault detection – experimental setup with the Bi-i V2 system

Experimental setup and the algorithmic solution: The inspection takes place on the machine that cuts a roll of stripe into labels (Fig. 8). Since the labels are moving fast, the time for integration on the image sensor must be short in order to avoid blurring. Because of the short shutter period very strong lighting is needed, which is solved by the use of hundreds of LED's driven by short current pulses synchronized with the image acquisition. The visual inspection system first takes a reference image from a perfect label and calculates different characteristic features. During the inspection process it takes an image from the label under test and compares it to the reference image.

This comparison is a difficult task for a number of reasons:

- The labels run fast and the inspection cannot slow down the manufacturing.
- The labels may be distorted which is a normal behavior when they are pulled through the machine. Within certain defined limits the system compensates for distortion and the label will not be marked as faulty despite the fact that the image significantly differs from the reference image.
- The fine texture of the woven label appears as noise in the images. Faults below this noise level must be also detected (if they are large enough). A grease spot is a good example for this situation.

The label inspection algorithm consists of the following steps:

- an image is acquired and the vertical position of the label is first calculated
- the actual positions of some characteristic patterns are identified
- the distortion of the label is corrected based on the characteristic positions

- the image is compared to the reference image resulting in an error image
- applying several adaptive threshold levels binary error images are generated
- for a threshold above the noise level one or a few active pixels on this image results an error signal
- for lower threshold levels several active pixels must be grouped together to indicate an error
- long but thin horizontal or vertical groups are also typical and indicate an error caused by missing or broken fibers

Results: The current implementation of the label inspection algorithm runs on the Bi-i v2. The resolution is about 0.3 mm/pixel, thus a typical 5 cm by 10 cm label is represented as an image of 55.500 pixels. The inspection algorithm runs on a label within 25 milliseconds. In other words 40 images are processed in a second; i.e. the labels can be inspected at 4 m/s. This image processing capacity fully satisfies the requirements of the narrow fabric industry.

4. Conclusions and Future Work

The Bi-i standalone cellular vision system has been introduced. We have discussed the underlying chip and system level architectures, algorithmic structures and various implementations. Several ultra high frame rate application examples have been discussed that highlight the most competitive feature of this computing platform with integrated sensing-processing-actuating capability.

Future work and upcoming papers will address the employment of the Bi-i system in more sophisticated out-door computing scenarios such as image fusion, multi-target tracking, visual search and navigation.

Acknowledgements

Thanks are due to the entire research and development team at AnaLogic-Computers Ltd. and Eutecus Inc. for the on-going work related to the Bi-i system and computational infrastructures ([18], [19]). The authors also would like to acknowledge the support for this research through the Hungarian National Research and Development Program, “TeleSense” (NKFP 035/02/2001), “Robotszempár” (IKTA-4, OMF-00209/2002), and the support received from the MDA under the contract No. HQ000604C7098, and the Human Frontier Science Program (HFSP) Young Investigators' Grant awarded in 2003.

References

- [1] L. O. Chua, "CNN: a Vision of Complexity ", *Int. J. of Bifurcation & Chaos*, Vol. 7, No. 10, pp. 2219-2425, 1997.
- [2] T. Roska and L. O. Chua, "The CNN Universal Machine", *IEEE Trans. on Circ. & Syst.*, Vol. 40, pp. 163-173, 1993.
- [3] F. S. Werblin, T. Roska, and L. O. Chua, "The Analogic CNN Universal Machine as a Bionic Eye", *International Journal of Circuit Theory and Applications*, Vol. 23, pp. 541-569, 1995.
- [4] B. Roska and F. S. Werblin, "Vertical Interactions Across Ten Parallel Stacked Representations in Mammalian Retina", *Nature*, 410 pp. 583-587, 2001.
- [5] Cs. Rekeczky, B. Roska, E. Nemeth, and F. Werblin, "The Network Behind Spatio-temporal Patterns: Building Low-complexity Retinal Models in CNN Based on Morphology, Pharmacology and Physiology", *International Journal of Circuit Theory and Applications*, Vol. 29, pp. 197-239, March-April 2001.
- [6] D. Bálya, B. Roska, T. Roska, and F. Werblin, "A CNN Framework for Modeling Parallel Processing in a Mammalian Retina", *International Journal of Circuit Theory and Applications*, Vol. 30, pp. 363-393, 2002.
- [7] S. Espejo, R. Carmona, R. Domínguez-Castro, and A. Rodríguez-Vázquez, "CNN Universal Chip in CMOS Technology", *International Journal of Circuit Theory and Applications*, Vol. 24, pp. 93-111, 1996.
- [8] S. Espejo, R. Domínguez-Castro, G. Liñán, Á. Rodríguez-Vázquez, "A 64x64 CNN Universal Chip with Analog and Digital I/O", in *Proc. ICECS'98*, pp. 203-206, Lisbon 1998.
- [9] G. Liñán, R. Domínguez-Castro, S. Espejo, A. Rodríguez-Vázquez, "ACE16k: A Programmable Focal Plane Vision Processor with 128 x 128 Resolution", *ECCTD '01 - European Conference on Circuit Theory and Design*, pp. 345-348, August 28-31, Espoo, Finland, 2001.
- [10] Cs. Rekeczky, T. Roska, and A. Ushida, "CNN-based Difference-controlled Adaptive Nonlinear Image Filters", *International Journal of Circuit Theory and Applications*, Vol. 26, pp. 375-423, July-August 1998.
- [11] K. R. Crouse and L. O. Chua, "Methods for Image Processing in Cellular Neural Networks: A Tutorial", *IEEE Trans. on Circuits and Systems*, Vol. 42, No. 10, pp. 583-601, October 1995.
- [12] B. E. Shi, T. Roska and L.O. Chua, "Estimating Optical Flow With Cellular Neural Networks," *Int. J. Circuit Theory and Its Application*, vol. 26, pp. 343-364, 1998.
- [13] Á. Zarándy, R. Domínguez-Castro, and S. Espejo, "Ultra-high Frame Rate Focal Plane Image Sensor and Processor", *IEEE Sensor Journal*, Vol. 2, No. 6 pp.:559-565, December 2002
- [14] A. Radványi, "On the Rectangular Grid Representation of General CNNs", *6th IEEE International Workshop on Cellular Neural Networks and their Applications CNNA 2000*, pp. 387-394, Catania, May 2000.
- [15] N. H. Packard and S. Wolfram, "Two-dimensional Cellular Automata", *J. of Statistical Physics*, Vol. 38, pp. 126-171, 1985.

- [16] A. R. Vázquez, G. Linan-Cembrano, L. Carranza, E. Roca-Moreno, R. Carmona-Galan, F. Jimenez-Garrido, R. Dominguez-Castro, and S. Espejo-Meana, "The 3rd Generation of Mixed-signal SIMD-CNN ACE Chips Toward VSoCs", IEEE Transaction on Circuits and Systems, Vol. 51, No 5, May 2004.
- [17] T. Roska, "CNN Technology / Cellular Wave Computers for Brain-like Spatio-temporal Computing", the companion paper in this issue.
- [18] AnaLogic Computers Ltd: <http://www.analogic-computers.com/>, Budapest 2005.
- [19] Eutecus Inc: <http://www.eutecus.com/>, Berkeley 2005.
- [20] Cs. Rekeczky, I. Szatmári, D. Bálya, G. Tímár, and Á. Zarándy, "Cellular Multi-Adaptive Analogic Architecture: A Computational Framework For UAV Applications", IEEE Transactions on Circuits and Systems I, Vol. 51, No. 5, pp. 864-884, 2004.