24

# Biologically Inspired Vision Architectures: a Software/Hardware Perspective

Francesco S. Fabiano, Antonio Gentile, Marco La Cascia
and Roberto Pirrone
*Dipartimento di Ingegneria Informatica – Università di Palermo*
*Italy*

## 1. Introduction

Even tough the field of computer vision has seen huge improvement in the last few decades, computer vision systems still lack, in most cases, the efficiency of biological vision systems. In fact biological vision systems routinely accomplish complex visual tasks such as object recognition, obstacle avoidance, and target tracking, which continue to challenge artificial systems. The study of biological vision system remains a strong cue for the design of devices exhibiting intelligent behaviour in visually sensed environments but current artificial systems are vastly different from biological ones for various reasons. First of all, biologically inspired vision architectures, which are continuous-time and parallel in nature, do not map well onto conventional processors, which are discrete-time and serial. Moreover, the neurobiological representations of visual modalities like colour, shape, depth, and motion are quite different from those usually employed by conventional computer vision systems. Despite these inherent difficulties in the last decade several biologically motivated vision techniques have been proposed to accomplish common tasks. For example Siagian & Itti [14] developed an algorithm to compute the gist of a scene as a low-dimensional signature of an image, in the form of an 80-dimensional feature vector that summarizes the entire scene. The same authors also developed a biologically-inspired technique for face detection [13]. Interesting results have also been reported in generic object recognition and classification (see for example [15] [16] [12] [11]). Also on the sensor side the biological vision systems are amazingly efficient in terms of speed, robustness and accuracy. In natural systems visual information processing starts at the retina where the light intensity is converted into electrical signals through cones and rods. In the outer layers of the retina the photoreceptors are connected to the horizontal and bipolar cells. The horizontal cells produce a spatially smoothed version of the incoming signal while the bipolar cells are sensitive to the edges in the image. Signals output from the cells are then used for higher level processing. Several architecture have been proposed to mimic in part the biological system and to extract information ranging from low to high level. For example Higgins [10] proposed a sensor able to perform an elementary visual motion detector. Other researchers proposed sensor to detect mid-level image features like corners or junctions [4] or even to perform higher level tasks such as tracking [6] or texture classification [5]. Robotics represents a typical field of application for hardware implementations of biologically inspired vision architectures.



Source: Vision Systems: Applications, ISBN 978-3-902613-01-1
Edited by: Goro Obinata and Ashish Dutta, pp. 608, I-Tech, Vienna, Austria, June 2007

Robot vision routines such as self localization, or 3D perception via calibrated cameras require large computing capabilities. Autonomous robot platforms have limited space to dedicate to such high level tasks because on board computers are busy most the time with motor control, and sensorial data acquisition. Even more limited embedded hardware is available on small wheeled robots for which almost all sensory computation is delegated to remote machines. Also in the case of robots equipped with onboard computer, most processing focuses on motion control, and low level sensorial data elaboration while heavy computer vision tasks, like image segmentation and object recognition, are performed in background, via fast connections to a host computer. Emerging gigascale integration technologies offer the opportunity to explore alternative approaches to domain specific computing architectures that can deliver a significant boost to on-board computing when implemented in embedded, reconfigurable devices. This paper describes the mapping of low level feature extraction on a reconfigurable platform based on the Georgia Tech SIMD Pixel Processor (SIMPil).

In particular, an adaptation of the Boundary webs Extractor (BWE) has been implemented on SIMPil exploiting the large amount of data parallelism inherently present in this application. The BWE [1] is derived from the original Grossberg's Boundary Contour System (BCS) and extracts a dense map of iso-luminance contours from the input image. This map contains actual edges along with a compact representation of local surface shading, and it is useful for high level vision tasks like Shape-From-Stereo. The Fast Boundary Web Extraction (fBWE) algorithm has been implemented in fixed point as a feed-forward processing pipeline thus avoiding BWE feedback loop, and achieving a considerable speed-up when compared against the standard algorithm. Application components and their mapping details are provided in this contribution along with a detailed analysis of their performance. Results are shown that illustrate the significant gain over a sequential implementation, and most importantly, the execution times in the order of 170 μsec for a 256000 pixel image. These results allow ample room for real-time processing of typical subsequent tasks in a complete robot vision system. The rest of this chapter is organized as follows. Section II introduces the Georgia Tech SIMPil architecture, and implementation efforts on FPGA. Section III provides some remarks on the original Grossberg's BCS, and its derived BWE model. In section IV the fBWE system is described, and its mapping onto SIMPil detailed. Section V reports extensive experiments with the fBWE compared with the BWE results, while in section VI some conclusions are drawn.

## 2. SIMPil FPGA implementation

The GeorgiaTech SIMD Pixel Processor (SIMPil) architecture consists of a mesh of SIMD processors on top of which an array of image sensors is integrated [8] [7]. A diagram for a 16-bit implementation is illustrated in Figure 1. Each processing element includes a RISC load/store datapath plus an interface to a 4×4 sensor subarray. A 16-bit datapath has been implemented which includes a 32-bit multiply-accumulator unit, a 16 word register file, and 64 words of local memory (the ISA allows for up to 256 words). The SIMD execution model allows the entire image projected on many PEs to be acquired in a single cycle. Large arrays of SIMPil PEs can be simulated using the SIMPil Simulator, an instruction level simulator. Early prototyping efforts have proved the feasibility of direct coupling of a simple processing core with a sensor device [3]. A 16 bit prototype of a SIMPil PE was designed in 0.8 μm CMOS process and fabricated through MOSIS. A 4096 PE target system has been

used in the simulations. This system is capable of delivering a peak throughput of about 5 Tops/sec in a monolithic device, enabling image and video processing applications that are currently unapproachable using today's portable DSP technology. The SIMPil architecture is designed for image and video processing applications. In general, this class of applications is very computational intensive and requires high throughput to handle the massive data flow in real-time. However, these applications are also characterized by a large degree of data parallelism, which is maximally exploited by focal plane processing. Image frames are available simultaneously at each PE in the system, while retaining their spatial correlation. Image streams can be therefore processed at frame rate, with only nominal amount of memory required at each PE [8]. The performance and efficiency of the SIMPil have been tested on a large application suite that spans the target workload.

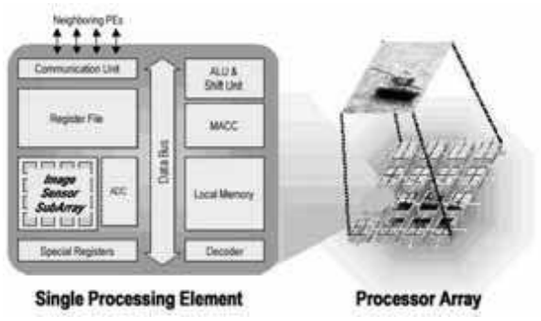



Figure 1. The SIMPiL architecture

For the SIMPil processing element, an application suite is selected from the DARPA Image Understanding suite [17]. These applications, listed in Table 1, are expressed in SIMPil assembly language, and executed using an instruction level simulator, SIMPilSim which provides various execution statistics. This simulator provides execution statistics including dynamic instruction frequency, operand size profiles, PE utilization, and PE memory usage. All applications are executed on a simulated 4096 processing element system with 16 pixels mapped to each PE for an aggregate 256×256 image size. All applications run well within real-time frame-rates and exhibit large system utilization figures (90% or more for most application). Details can be found in [8]. To bring SIMPil performance onto robot platform, a reconfigurable platform based on FPGA devices is being developed. This platform uses a parameterized SIMPil core (SIMPil-K) described in the VHDL hardware description language. The SIMPil-K platform is an array of Processing Elements (PE) and interconnection registers which can be configured to fit any FPGA device at hand. Figure 2 shows the high-level functional schema of a 4×4 SIMPil-K array and its NEWS interconnection network. Each NEWS register supports communication among a particular node (i.e. PE) and its north and west neighbours. By replicating this model, a NEWS (North, East, West, South) network is obtained, with every node connected to its four neighbours. SIMPil-K receives an instructions stream through a dedicated input port. The instruction stream is then broadcast to each PE. To upload and download image data, SIMPil-K uses a boundary I/O mechanism, supported by its boundary nodes (i.e. PEs laid on its East/West edge): every east-edge node uploads a K-bit data word from its boundary-input port to the general purpose register file; every west-edge node downloads a K-bit data word from its

register file to the boundary output port. An upload/download operation (one word per node) takes only one clock cycle. Both boundary input and output operations are enabled by a single instruction, XFERB. When a NEWS transfer instruction arrives, it needs only one clock cycle to transfer the data word from each node to a neighbour one, in a specified direction. The SIMPil-K platform can be reconfigured by varying a number of architectural parameters, as detailed in Table 2. This allows for experimentation with a large set of different system configurations, which is instrumental to determine the appropriate system characteristics for each application environment AW and RAW parameters set the address space of register file and memory, respectively. PPE specifies the number of image pixels mapped to each PE. The Influence parameter toggle between a fixed instruction width (24 bit) and a variable one (8+K bits). The interface of a processing element is depicted in Figure 3, below. There are two input ports for clock signals, a reset input port and the instruction stream port. NEWS transfers are carried through the three bidirectional dedicated ports (NEWS ports) which drive three NEWS buses, namely the North/West Bus, East Bus and South Bus.

| **Image Transforms** | **Image Enhancement** |
|---|---|
| Discrete Fourier Transform | Intensity Level Slicing |
| Discrete Cosine Transform | Convolution |
| Discrete Wavelet Transform | Magnification |
| Image Rotation | Median Filtering |
| **Image/Video Compression** | **Image Analysis** |
| Quantization | Morphological Processing |
| Vector Quantization | Region Representation |
| Entropy Coding | Region Autofocus |
| JPEG Compression | K-means Classification |
| Motion Estimation | |
| MPEG Compression | |

Table 1. SIMPil Application Suite

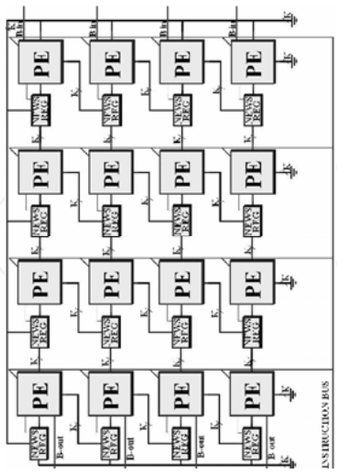



Figure 2. K-bit 4-by-4 SIMPil-K array

Boundary data input and output are carried through the two dedicated boundary ports. The processing element parameterized architecture is described in Figure 4. There are four communication buses shared by the functional units. All functional units can be reconfigured based on the datapath width selected. A single PE can perform integer operations on K-bits. Dedicated barrel shift unit and multiply-accumulate unit are instrumental to speed-up most image processing kernels. The Sleep Unit verifies and updates the node activity state, thus allowing execution flow control based on each PE local data. The SIMPil-K system has been simulated and synthesized on FPGA; synthesis statistics about employed resources has been generated and analyzed. Figure 5 shows resources use percentage achieved by implementing several 16-bit SIMPil-K versions on an eight million gates FPGA: particularly, 2-by-2, 4-by-4 and 8-by-8 16-bit SIMPil-K arrays have a resources use percentage respectively of 3.3%, 13.3%, and 53.3%.

| Parameter | Function | Values | Constr. | Def. |
|---|---|---|---|---|
| $K$ | Word Width | {8, 16, 32, 64} | – | 16 |
| $X$ | Array Columns | $X \in \mathrm{N}$ | $X, Y = 2^j, j \in \mathrm{Z}$ | 4 |
| $Y$ | Array Rows Register | $Y \in \mathrm{N}$ | | 4 |
| $AW$ | File Address Width | $AW \in N \cap [1, 16]$ | $I = \text{off} \rightarrow AW \leq 4$ $I = \text{on} \rightarrow AW \leq (K/4)$ | 4 |
| $RAW$ | Local RAM Address Width | $RAW \in N \cap [1, 16\}$ | $RAW \leq (K/4)$ | 4 |
| $PPE$ | Pixel per Processing | $PPE \in N$ | $PPE = p^2, p \in N, PPE \leq 2^K$ | 8 |
| Influence (I) | Instructions Format Change Enable | $I \in$ [on, off] | – | off |

Table 2. SIMPil-K Architectural Parameters

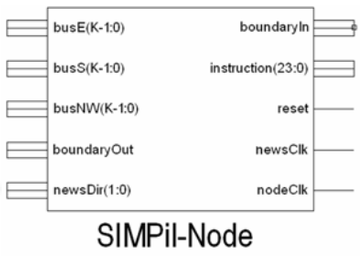



Figure 3. The Processing Element Black Box

## 3. The Boundary Webs Extractor

The original BCS architecture was proposed by Grossberg and Mingolla [9] as a neural model, aimed to explain some psychological findings about perceptual grouping of contours in vision: it was part of a more complex theory regarding human perception of shapes and

colors. In this formulation, the BCS is a multi-layer recurrent network trained using a competitive/cooperative scheme until an equilibrium state is reached. BCS units have dynamic activations that are expressed using differential equations with respect to time. The network takes the input from a gray-level image, with a lattice of receptive fields computing local contrast in small areas. Output is provided as a 2D map of vectors, with the same spatial displacement of the input receptive fields, which are called boundary webs, and describe changes in brightness over the image. A boundary web is locally oriented along a constant brightness line, meaning that image contrast changes along the orthogonal direction. The amplitude of each boundary web is related to the strength of the local contrast. Boundary webs form a piecewise linear approximation of all image contours, while they follow iso-luminance paths inside smoothly shaded surfaces: consequently, they can be regarded as a compact description of image shading. a typical BCS analysis is described in Figure 7(b), while Figure 6 reports an outline of the BCS architecture. The network consists of an input stage used to collect contrast information, the so called OC Filter, and of three layers: *Competition I, Competition II* and *Cooperation*. The OC Filter is used to collect local image contrast along different directions without taking into account contrast orientation.

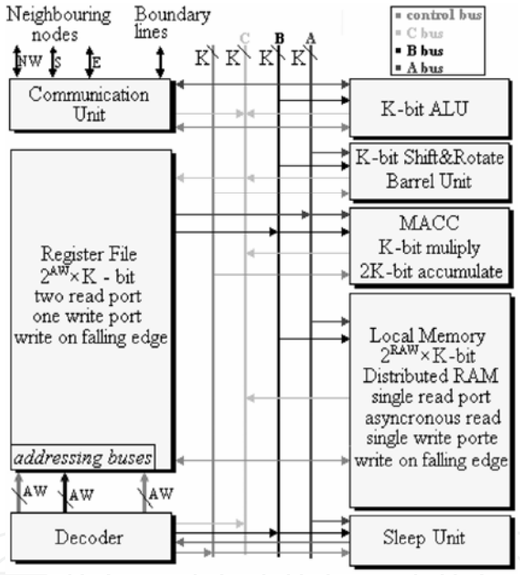



Figure 4. Processing Element K-bit Datapath

All subsequent layers are arranged as a lattice of complex cells, with the same spatial displacement of the receptive fields. Each cell in the lattice has a pool of activation values which are related to the various contrast directions. The first two layers are competitive ones, and their purpose is to refine locally detected contours. The third (output) layer performs long range cooperation between neighboring cells in order to complete extended contours across the image. Finally, the feedback loop is again competitive, and is connected to the first layer in order to enforce winner cells activations. In the OC Filter circular receptive fields at position *(i,j)* sum up input pixels from a squared sub-image $\mathbf{S} = [S_{pq}]$ in

two symmetric halves $L_{ijk}$ and $R_{ijk}$ defined for each mask at the $k$-th orientation. Assuming that $[x]^+ = \max(x, 0)$, the resulting activation at position $(i, j)$ and orientation $k$ is:

$$J_{ijk} = \frac{[U_{ijk} - \alpha V_{ijk}]^+ + [V_{ijk} - \alpha U_{ijk}]^+}{1 + \beta(V_{ijk} + U_{ijk})} \quad (1)$$

where $U_{ijk}$ and $V_{ijk}$ are the summed input in the mask's halves, while $\alpha$ and $\beta$ are suitable constants. The first competitive layer enforces local winner activations via the feedback signal and the input from the OC Filter, while tends to decrease activation in neighboring cells with the same orientation. In case of strong aligned activations induced by image contours, the aim of the first competitive stage is to reduce the activation diffusion beyond contours endpoints.

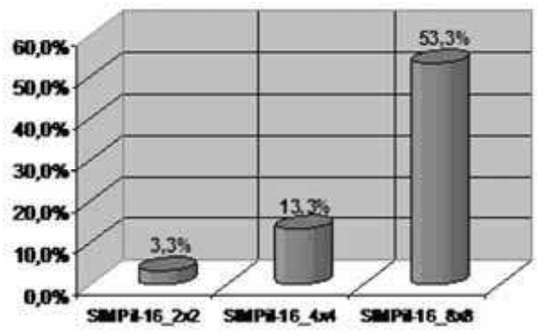



Figure 5. Used resources on eight million gates FPGA

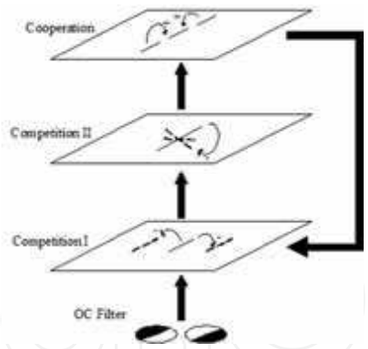



Figure 6. The BCS architecture

This effect results in the illusory contours completion phenomenon which is commonly observed in human perception. Activation laws are, in general, differential equations with respect to time, but in the BCS computational model they are computed at equilibrium ($d/dt = 0$). In the case of the Competition I layer the dynamic activation rule is:

$$\frac{dw_{ijk}}{dt} = w_{ijk} + I + BJ_{ijk} + v_{ijk} - Bw_{ijk} \sum_{(p,q)} J_{pqk} A_{pqij} \quad (2)$$

and the equilibrium activation $w_{ijk}$ for each cell in this stage is computed as:

$$w_{ijk} = \frac{I + BJ_{ijk} + v_{ijk}}{1 + B\sum_{(p,q)} J_{pqk}A_{pqij}} \tag{3}$$

where $v_{ijk}$ is the feedback signal, $A_{pqij}$ are the coefficient of a small kernel with cylindrical shape, while $I$ and $B$ are suitable constants. In following equations capital letters without indexes are constant values used to tune the model. The second competitive stage performs competition among orientations inside the same cell: this is a local contour refinement mechanism which will be enhanced by the cooperative stage. The activation law has the following form:

$$y_{ijk} = \frac{C[w_{ijk} - w_{ijK}]^+}{D + \sum_{m=1}^{n}[w_{ijm} - w_{ijM}]^+} \tag{4}$$

where capital indexes are referred to orthogonal direction with respect to the current one. The cooperative stage performs long range cooperation between cells with the same orientation that are displaced in a wide neighborhood. In this way long contours completion is enabled. Considering the vector d connecting the position $(i, j)$ with a generic neighbor $(p,q)$, the following quantities can be defined $N_{pqij} = |\mathrm{d}|$ and $Q_{pqij} = \angle \mathrm{d}$, while the cooperative activation law is:

$$\begin{aligned} z_{ijk} = {} & g\Big(\sum_{(p,q,r)} (y_{pqr} - y_{pqR})[G^{(r,k)}_{pqij}]^+\Big) \\ & + g\Big(\sum_{(p,q,r)} (y_{pqr} - y_{pqR})[-G^{(r,k)}_{pqij}]^+\Big) \end{aligned} \tag{5}$$

where:

$$\begin{aligned} g(x) &= \frac{H[x]^+}{K + [x]^+}, \\ G^{(r,k)}_{pqij} &= \exp(-2(N_{pqij}/P - 1)^2) \cdot \\ &\quad |\cos(Q_{pqij} - r)|^R \cos(Q_{pqij} - k)^T \end{aligned}$$

This very complex kernel has the form of two elongated blobs aligned with the orientation $k$, and exponentially decreasing towards 0. In particular, $P$ represents the optimal distance from the cooperative cell at which maximum input activation is collected. Finally, feedback is provided from the cooperative stage to the first competitive one, in order to enforce those activations that are aligned with emergent contours and decrease spurious ones. The form of the feedback signal is:

$$v_{ijk} = \frac{L[z_{ijk} - M]^+}{1 + L\sum_{p,q}[z_{pqk} - M]^+ W_{pqij}} \tag{6}$$

where $W_{pqij}$ are the coefficient of a small cylinder shaped kernel. BCS provides a compact description of image shading at selectable resolution levels: shading, in turn, can be used to perform shape estimation, while boundary webs can be used as low level features for contour extraction, alignment, or stereo matching. Possible uses of BCS have been explored

by some of the authors resulting in a software implementation of the BCS, called Boundary Web Extractor (BWE) which has been used as a low level feature extraction module in different vision systems. In particular, a neural shape estimation model has been proposed [1] coupling BWE analysis with a backpropagation network trained to classify BWE patterns as belonging to superquadrics surface patches. Input image surfaces are processed by BWE, and the BWE output pertaining to different ROIs is modeled in terms of superquadrics.
Another approach [2] performs BWE analysis on stereo couples. Input images are analyzed both with standard correlation operator over pixels intensities, and with BWE as a supplementary feature. Candidates points are labeled using a measure of the matching probability with respect to both the preprocessing operators. Finally, a relaxation labeling algorithm provides matches for almost all points in the image, and disparities are obtained.
The high resolution achievable by the BWE analysis enables dense depth maps. The main objective of BWE is to perform local brightness gradient estimation, without taking into account the support for perception theories. In this perspective BWE has been slightly modified with respect to BCS, to obtain sharp contrast estimation and emergent contours alignment. In particular, $N$ couples of dually oriented Gabor masks have been used as receptive fields to obtain n activation values discarding, for each couple, the mask providing negative output. The resulting OC Filter is described by the following equation:

$$J_{ijk} = [U_{ijk}]^+ + [V_{ijk}]^+ \tag{7}$$

where $U_{ijk}$ and $V_{ijk}$ are the outputs of two dual Gabor masks. The generic Gabor filter has been selected in our implementation with a width $w$ equal
to 8 pixels, $2N = 24$. The filter equation is:

$$\begin{aligned} M_{ijk} = &\ \alpha e^{-\beta(\gamma B^2 + C^2)} \sin(\delta C) \\ B = &\ (w-p)\cos(2k\pi/N) - (q-s)\sin(2k\pi/N) \\ C = &\ (w-p)\sin(2k\pi/N) + (q-s)\cos(2k\pi/N) \end{aligned} \tag{8}$$

Here $s$ is the application step of the masks; the α...δ parameters have been heuristically tuned. The kernel in eqs. (3) and (6) have been selected with gaussian shape, and the subtractive term in the exponential part of $G^{(r,k)}_{pqij}$ kernel has been suppressed, and all constant values in the equations have been suitably tuned. To ensure the kernel to be symmetric, its central value has been forced to be 0 in order to avoid the exponential function to give a positive value when $N_{pqij} \equiv 0$. Finally, we can give a formulation of the BWE structure as a 3D matrix containing, at each location ($i$, $j$), $2N$ activation values belonging to a star of vectors.

$$\begin{aligned} \mathbf{BW} &= [\mathbf{B}_{ij}] \quad i,j = 1,\ldots,M \\ \mathbf{B}_{ij} &= \{\mathbf{b}_{ijk}\} \quad k = 1,\ldots,2N \end{aligned} \tag{9}$$

Each vector represents the value of the image contrast along the orthogonal direction with respect to its phase. As a consequence of the modified OC Filter behaviour, the location $\mathbf{B}_{ij}$ of the **BW** matrix contains $N$ couples, each of them having a null vector that corresponds to the negative output of the filter with at same orientation.

$$\mathbf{b}_{ijk} = b_{ijk} e^{j\theta_{ijk}}, \ \ b_{ijk} = \max\left(|\mathbf{b}_{ijk}|, 0\right), \ \ \theta_{ijk} = k\pi/N \tag{10}$$

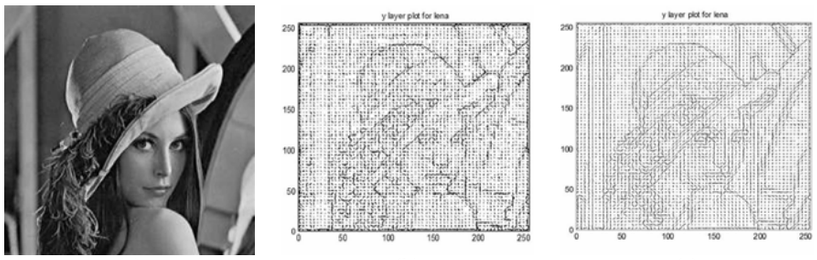
For computer vision purposes the average boundary webs are noticeable because they provide a single estimation of the local image contrast at each spatial location, both as intensity and direction. The average process is computed using a suitable average function $f_{av}$:

$$\begin{aligned} &\mathbf{A}_{BW} = [\mathbf{a}_{ij}], \;\; i,j = 1,\ldots,M \\ &\forall i,j \;\; \mathbf{a}_{ij} = \; a_{ij}e^{j\theta_{ij}} = \mathrm{f}_{av}\left(\mathbf{B}_{ij}\right) \end{aligned} \tag{11}$$

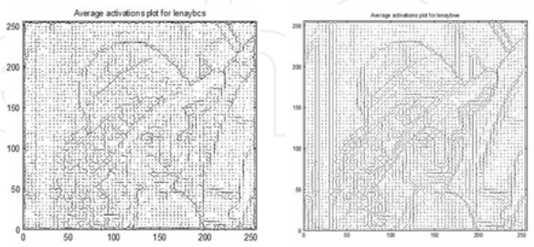
The average function can be selected according to several criteria: the maximum value or the vector sum of all the elements at each location; we selected a form of $f_{av}$ that weights each intensity with the cosine of the angle between the phase value and a mean phase angle, obtained weighting each phase with the respective intensity.

$$\begin{aligned} &\mathrm{f}_{av}: \;\; \theta_{ij} = k_M\pi/N, \;\; k_M = \sum_{k=1}^{2N} b_{ijk}k \Big/ \sum_{k=1}^{2N} b_{ijk} \\ &a_{ij} = \frac{\sum\limits_{k=1}^{2N} \mathrm{abs}\left(\cos\left(\theta_{ijk} - \theta_{ij}\right)\right) b_{ijk}}{\sum\limits_{k=1}^{2N} \mathrm{abs}\left(\cos\left(\theta_{ijk} - \theta_{ij}\right)\right)} \end{aligned} \tag{12}$$

Figure 7 makes a comparison between the original BCS and BWE both for the actual output, and for the average one.




(a) (b) (c)




(d) (e)

Figure 7. Comparison between BCS ((b),(d)) and BWE ((c),(e))

## 4. The fBWE system

The main idea about the fBWE implementation is to design a massively parallel algorithm that should be robust with respect to noise while producing an output as similar as possible to the true BWE architecture. The main performance drawbacks of the BWE network are the presence of a feedback loop aimed to put the whole system in a steady state, and the use of floating point calculations. The fBWE system is a feed-forward elaboration pipeline that is completely implemented using 16-bit integer maths, according to SIMPil-K requirements. In Figure 8 the fBWE pipeline is shown. The fBWE architecture relies on the cascade of the OC Filter, and a competitive-cooperative pipeline. The SIMPil-K configuration we used, is made of 32×32 PEs with a PPE equal to 64, that is each sub-image is 8×8 pixels wide.

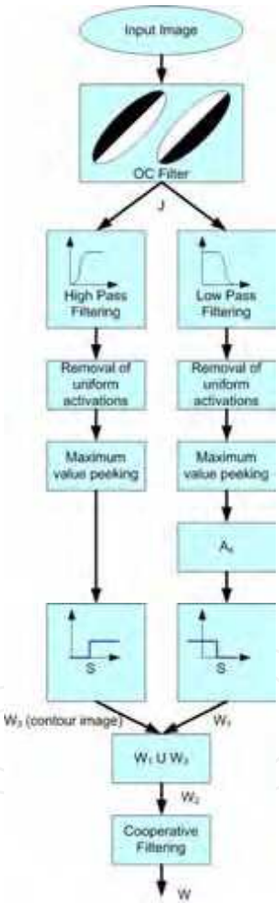



Figure 8. The fBWE pipeline

The whole process has been applied to 256×256 images, and $M$ = 64 so there is a 4 pixels overlapping along each direction between two adjacent neighborhoods. Gabor masks in the

OC Filter have been implemented using equation (8), and have been provided to the PE array as a suitable gray level image. The original floating point values obtained for the weights have been approximated to 8-bit integer values, and the minimum value has been added to each of them to obtain a correct dynamics in the range [0,....,255]. The set of Gabor masks is depicted in Figure 9. The same mask is loaded into all the PEs in one column.

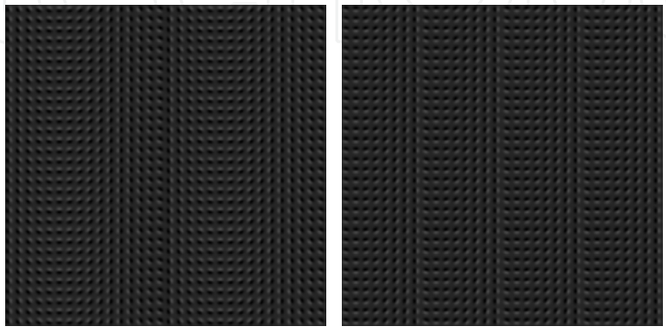


Figure 9. Arrangement of the Gabor masks for the PEs

Each row of the first image contains only 16 different orientations repeated twice, while the second one depicts the last 8 orientations repeated four times. At loading time, the offset is subtracted from each Pixel Register in the PE to correct the weights. After loading the input image the true filtering starts. The R15 register of each PE contains the correct value for the orientation $k$ in order to store the result in the correct position after each filtering step. Due to the overlapping, each mask is used to convolve four neighborhoods shifting only one half of a sub-image between two PEs at each step, according to the scheme West-North-East-South. Finally, the Gabor masks image is shifted in the West direction by 8 pixels starting again the filtering cycle. The same procedure is adopted for the second Gabor masks image, but the filtering cycle is iterated only 8 times. After the filtering phase each PE contains four adjacent locations each containing $N$ non null orientations due to the application of equation (7). The OC Filter output is quit precise in the determination of the orientations, but it suffers from its *locality*. Contours are not perfectly aligned, and they tend to double along a direction due to the activations present in couples of overlapped regions which intersect the same contour line. The competitive-cooperative pipeline tends to eliminate these problems without the use of a feedback scheme. Here the outputs of the OC Filter are grouped as $N$ *orientation* images 64×64 pixels wide. The pipeline is split into two parallel branches: at the first step each orientation image is processed with a 3×3 high pass filter in the left branch, and a median filter of the same size in the right one. The left processing is aimed to enrich details, and to strengthen the contours, while the median filter is a form of blurring intended force close orientations to align thus correcting the OC Filter spurious outputs. The implementation of these filters in SIMPil-K implies that each PE needs a frame of 12 values surrounding the ones stored in its local memory. So a suitable transfer routine has been set up to obtain these values form the 8-neighborhood surrounding the PE. The four filtered values are again stored in the PE's local memory. The next step in both the pipeline branches is the suppression of uniform activation values. When an image region insisting on the location $(i, j)$ exhibits a uniform luminance without perceivable contrast variation along any

direction the fBWE activations $b_{ijk}$ are almost of the same magnitude and a sort of little star is visualized in the output. To avoid this behaviour the uniform activations suppression acts according to the following rule:

$$\begin{aligned} \tilde{b}_{ij} - \hat{b}_{ij} \leqslant 0.2\tilde{b}_{ij} \Rightarrow \forall k\, b_{ijk} \equiv 0 \\ \tilde{b}_{ij} = \max_k(b_{ijk}) \\ \hat{b}_{ij} = \min_k(b_{ijk}) \end{aligned} \tag{13}$$

Here the threshold value of 0.8 has been selected on the basis of a trial and error process. After uniform activations suppression the maximum values $\tilde{b}^l_{ij}$ and $\tilde{b}^r_{ij}$ are selected at each location for the left and right branches thus obtaining two average boundary webs images, using $max(\blacksquare)$ in place of the averaging function $f_{av}$. High pass, and median filters give rise to extremely different dynamics in the two pipeline branches, so a gain element has been placed in the high pass branch to normalize these ranges. The gain factor has been determined as

$$A_s = \frac{\max_{ij}(\tilde{b}^r_{ij})}{\max_{ij}(\tilde{b}^l_{ij})} \tag{14}$$

In all our experiments $A_s$ assumed values between 6 and 7. Before the conjunction of the two branches with the union pixel by pixel of the left ($W_L$) and right ($W_R$) image, a sharp threshold $S$ has been applied in order to join exactly $W_L$ and $W_R$. The value of S has been selected as the 30% of the maximum activation in $W_L$, and all the values in $W_R$ that are over the value of $S$ are joined with all the values of $W_R$ that are beneath the same threshold. The joined image $W_J$ can be defined as $W_J = [(W_{J,ij}, k_{ij})]$ where for each location $(i, j)$ the amplitude, and the relative orientation value are defined. The last step is the cooperative filtering that generates the fBWE image $W$, and is aimed to enforce aligned neighboring activations.

An activation is enforced if its orientation is slightly different from the one of the location at the center of the filter mask, otherwise it is decreased. The generic weight $M_{pq}$ of the filter applied to the location $(i, j)$ is defined as:

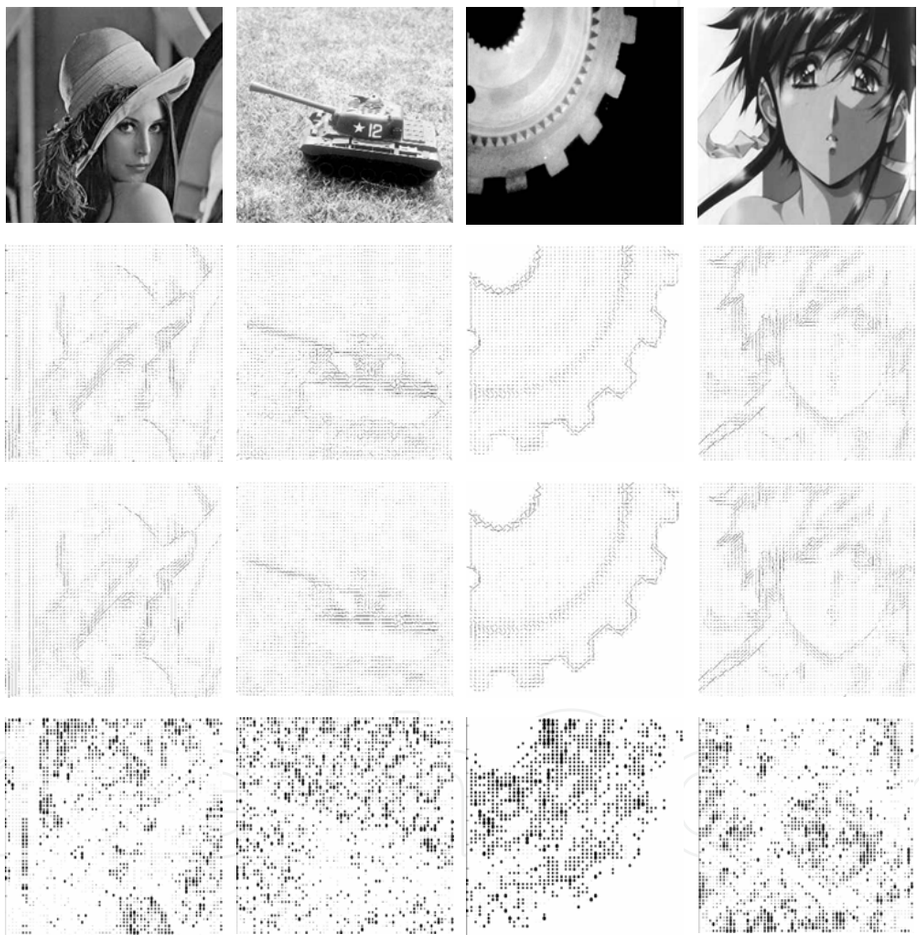
$$M_{pq} = \begin{cases} 1 - \frac{|k_{pq} - k_{ij}|}{N/2}, & \frac{|k_{pq} - k_{ij}|}{N/2} < N/2 \\ 1 - \frac{||k_{pq} - k_{ij}| - N|}{N/2}, & \text{otherwise} \end{cases} \tag{15}$$

Also in this case it is necessary for each PE to obtain 12 values from its eight neighbors.

## 5. Experimental Results

Several experiments have been conducted on a set of images with different pictorial features: real images with a lot of shading, highly textured images, high contrast ones, and artificial pictures with both high dynamics (like cartoons) and poor one (Kanizsa figures). In Figure 10 the BWE and fBWE images are reported along with a diagram of the local orientation differences $d_{ij} \triangleq k^{(BWE)}_{ij} - k^{(fBWE)}_{ij}$. It can be noticed that the two implementations are perceptually equivalent, and the major differences are present in the uniform brightness regions. In these parts of the image the BWE exhibits some small

residual activations due to the feedback based stabilization process, while the fBWE suppresses them at all. In the case of Kanizsa figures with a few well distinct gray levels (see Figure 11) the OC Filter alone performs better of the fBWE, so it has been selected as the system output. As regards the performance, the BWE execution time in our experiments ranges from 14.94 sec. in the case of Kanizsa figure to 68.54 sec. for the Lena and Tank images, while fBWE has a constant execution time of 0.168 msec. This is an obvious finding because the fBWE is a feed-forward architecture, while the BWE is not, and its convergence to a steady state depends on the input brightness structure.



a) Lena ( (b) Tank (c) Gear (d) Cartoon

Figure 10. Experimental results, from top to bottom: input image, BWE output, fBWE output, difference

(a) (b) (c)

Figure 11. Experimental results on a Kanizsa figure: (a) input image, (b) fBWE output, (c) OC Filter output

## 6. Conclusion

A Fast Boundary Web Extraction (fBWE) algorithm was presented in this paper as a fixed-point, data parallel implementation of the BWE. fBWE was mapped on SIMPil-K reconfigurable FPGA based platform.

Application components and their mapping details were provided along with a detailed analysis of their performance. Experimental results illustrate the significant gain achieved over the traditional BWE, with execution times allowing ample room for real-time processing of typical subsequent tasks in a complete robot vision system. Experimental results on an extensive data set illustrate the significant gain achieved over the traditional BWE implementation. Execution times are in the order of 170 μsec for a 256000 pixel image, thus allowing ample room for real-time processing of typical subsequent tasks in a complete robot vision system.

## 7, Reference

E. Ardizzone, A. Chella, R. Pirrone, and F. Sorbello. Recovering 3-D Form Features by a Connectionist Architecture. *Pattern Recognition Letters*, 15:77–85, 1994. [1]

E. Ardizzone, D. Molinelli, and R. Pirrone. A Fast Robust BCS Application to the Stereo Vision. In M. Marinaro and R. Tagliaferri, editors, *Neural Nets WIRN Vietri-95 Proceedings of the 7th Italian Workshop on Neural Nets*, pages 215–225, Vietri Sul Mare (SA), Italy, 1995. World Scientific Pu., Singapore. [2]

H.H. Cat, A. Gentile, J.C. Eble, M. Lee, O. Vendier, Y.J. Joo, D.S. Wills, M. Brooke, N.M. Jokerst, and A.S. Brown. SIMPil: An OE Integrated SIMD Architecture for Focal Plane Processing Applications. In *Proceedings of the Third IEEE International Conference on Massively Parallel Processing using Optical Interconnection* (MPPOI-96), pages 44–52, Maui Hawaii, USA, 1996. [3]

J. Van der Spiegel and M. Nishimura. Biologically inspired vision sensor for the detection of higher-level image features. In *2003 IEEE Conference on Electron Devices and Solid-State Circuits*, pages 11–16. IEEE Computer Society, Washington DC, USA, December 16-18 2003. [4]

R. Dominguez-Castro, S. Espejo, A. Rodriguez-Vazquez, R.A. Carmona, P. Foldesy, A. Zarandy, P. Szolgay, T. Sziranyi, and T. Roska. A 0.8-µm CMOS two-dimensional programmable mixed-signal focal-plane array processor with on-chip binary imaging and instructions storage. *IEEE Journal of Solid-State Circuits*, 32(7):1013–1026, July 1997. [5]

R. Etienne-Cummings, J. Van der Spiegel, P. Mueller, and Mao-Zhu Zhang. A Foveated Silicon Retina for Two-Dimensional Tracking. *IEEE Transactions on Circuits and Systems Part II: Express Briefs*, 47(6):504–517, June 2000. [6]

A. Gentile, S. Sander, L.M. Wills, and D.S. Wills. The impact of grain size on the eficiency of embedded SIMD image processing architectures. Journal of Parallel and Distributed Computing, 64:1318–1327, September 2004. [7]

A. Gentile and D.S. Wills. Portable Video Supercomputing. *IEEE Trans. on Computers*, 53(8):960–973, August 2004. [8]

S. Grossberg and E. Mingolla. Neural Dynamics of Perceptual Grouping: Textures, Boundaries and Emergent Segmentation. *Perception and Psychophysics*, 38:141–171, 1985. [9]

C.M. Higgins. Sensory architectures for biologically-inspired autonomous robotics. The Biological Bulletin, 200:235–242, April 2001. [10]

F.F. Li, R. Fergus, and P. Perona. Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. In *2004 Conference on Computer Vision and Pattern Recognition Workshop* (CVPRW'04), volume 12, page 178. IEEE Computer Society, Washington DC, USA, June 27 - July 02 2004. [11]

T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):411–426, 2007. [12]

C. Siagian and L. Itti. Biologically-Inspired Face Detection: Non-Brute- Force-Search Approach. *In 2004 Conference on Computer Vision and Pattern Recognition Workshop* (CVPRW'04), volume 5, page 62. IEEE Computer Society, Washington DC, USA, June 27 - July 02 2004. [13]

C. Siagian and L. Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):300–312, 2007. [14]

S. Ullman, E. Sali, and M. Vidal-Naquet. A Fragment-Based Approach to Object Representation and Classification. *In Proceedings of 4th International Workshop on Visual Form IWVF4*, volume LNCS 2059, pages 85–102, Capri, Italy, May 2001. Springer-Verlag, Heidelberg. [15]

P. Viola and M. Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. In 2001 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'01), volume 1, pages 511–518. IEEE Computer Society, Washington DC, USA, December 8-14 2001. [16]

C.C. Weems, E.M. Riseman, A.R. Hanson, and A. Rosenfield. The DARPA Image Understanding Benchmark for parallel Computers. *Journal of Parallel and Distributed Computing*, 11:1–24, 1991. [17]


**Vision Systems: Applications**

Edited by Goro Obinata and Ashish Dutta

ISBN 978-3-902613-01-1

Hard cover, 608 pages

**Publisher** I-Tech Education and Publishing

**Published online** 01, June, 2007

**Published in print edition** June, 2007

Computer Vision is the most important key in developing autonomous navigation systems for interaction with the environment. It also leads us to marvel at the functioning of our own vision system. In this book we have collected the latest applications of vision research from around the world. It contains both the conventional research areas like mobile robot navigation and map building, and more recent applications such as, micro vision, etc.The fist seven chapters contain the newer applications of vision like micro vision, grasping using vision, behavior based perception, inspection of railways and humanitarian demining. The later chapters deal with applications of vision in mobile robot navigation, camera calibration, object detection in vision search, map building, etc.

**How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Francesco S. Fabiano, Antonio Gentile, Marco La Cascia and Roberto Pirrone (2007). Biologically Inspired Vision Architectures: a Software/Hardware Perspective, Vision Systems: Applications, Goro Obinata and Ashish Dutta (Ed.), ISBN: 978-3-902613-01-1, InTech, Available from: http://www.intechopen.com/books/vision_systems_applications/biologically_inspired_vision_architectures__a_software_hardware_perspective

INTECH
open science | open minds

**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen