Real-time speckle image processing

Elías Todorovich, Ana Lucia Dai Pra, Lucia Isabel Passoni, Martín Vázquez, Ezequiel Cozzolino, Fernando Ferrara & Gery Bioul

Journal of Real-Time Image Processing

ISSN 1861-8200
Volume 11
Number 3

J Real-Time Image Proc (2016)
11:535-545
DOI 10.1007/s11554-013-0343-4

Springer

J Real-Time Image Proc (2016) 11:535–545
DOI 10.1007/s11554-013-0343-4

ORIGINAL RESEARCH PAPER

Real-time speckle image processing

Elías Todorovich · Ana Lucia Dai Pra · Lucia Isabel Passoni · Martín Vázquez · Ezequiel Cozzolino · Fernando Ferrara · Gery Bioul

Received: 25 October 2012 / Accepted: 22 March 2013 / Published online: 11 April 2013

Abstract The laser dynamic speckle is an optical phenomenon produced when a laser light is reflected from an illuminated surface undergoing some kind of activity. It allows a non-destructive process for the detection of activities that are not easily observable, such as seed viability, paint drying, bacterial activities, corrosion processes, food decomposition, fruit bruising, etc. The analysis of these processes in real time makes it possible to develop important practical applications of commercial, biological and technological interest. This paper presents a new digital system based on granular computing algorithms to characterize speckle dynamics within the time domain. The selected platform to evaluate the system is Field Programmable Gate Array (FPGA) technology. The obtained minimum clock periods and latencies enable speckle image processing with real-time constraints with a maximum throughput of about thousand 512 × 512 fps.

Keywords Dynamic speckle · Granular computing · Programmable logic · Real-time processing · Rough-fuzzy sets

E. Todorovich · M. Vázquez · E. Cozzolino · F. Ferrara · G. Bioul
Faculty of Engineering, FASTA University, Mar del Plata, Argentina
e-mail: gbioul@ufasta.edu.ar

E. Todorovich (✉) · M. Vázquez
Universidad Nacional del Centro de la Provincia de Buenos Aires, Campus Universitario, B7001BBO Tandil, Argentina
e-mail: etodorov@exa.unicen.edu.ar

A. L. Dai Pra · L. I. Passoni
Faculty of Engineering, Universidad Nacional de Mar del Plata, Mar del Plata, Argentina
e-mail: daipra@fi.mdp.edu.ar

1 Introduction

The laser speckle is an optical phenomenon that occurs when a reflected laser light from a lighted surface is shown as a granular pattern of high contrast. When there is some activity in the illuminated surface this speckle pattern evolves in time. The dynamics of the observed physical phenomenon can be assessed by one of the behaviors of interest called "boiling", because of its appearance, where the speckles disappear and reappear without any significant displacement. This behavior can be observed in different types of phenomena like biological (seed viability [4], bacterial activity [20], fruit bruising [14], etc.) or non-biological processes (drying of paints [7, 12], corrosion [9], etc.); the activity is given by the change of the sample properties due to diverse physical phenomena.

Captured sequences of laser speckle images must be processed offline via a general purpose computer to characterize the phenomena, using ad hoc-designed descriptors that analyze the behavior of every pixel across the sequence. A wide set of these descriptors can be found in [18]; many of them carry out processes in the time domain, such as the descriptor of Fujii [10], Generalized Differences [1], Fuzzy Granular Descriptor (FGD) [5], among others. Other descriptors carry out this task using frequency domain tools, like the High to Low Frequency ratio (HLR) [11], and Frequency Band Decomposition [19]. Others deal with the frequency–time domain, such as the Wavelet Entropy Descriptor [16].

These descriptors perform differently depending on the type of application. Cases belonging to the so-called biospeckle phenomena are quite difficult to characterize, particularly when it comes to detecting regions of interest in living specimens, such as regions of bruising in apples,

viability in corn seed or bacteria activity in swarm experiments.

One of the descriptors that showed greater efficacy in these types of applications, and in turn presented the lowest computational complexity, was the Fuzzy Granular Descriptor (FGD) based on a fuzzy framework, presented in [5] as Q_T. The FGD is neatly efficient for the segmentation of biospeckle regions compared with results deriving from time–frequency algorithms, like the Wavelet Entropy Descriptor, which has high computational complexity [6].

Given the real-time operation requirement, the aim of this work was to design a new version of the FGD algorithm using concepts that link the Rough set and the Fuzzy set theories, thus simplifying the process and reducing the computational complexity. The FGD algorithm was optimized for the design of a novel digital system that operates in real time. A Rough-Fuzzy Granular Descriptor (RFGD) algorithm was specifically designed for this implementation on the digital system.

The remainder of this article is organized as follows: Sect. 2 presents the bases of the dynamic speckle signal processing (DSSP) techniques, the FGD, and the transformation to RFGD; Sect. 3 explains the granular computing algorithm with an emphasis on memory requirements; Sect. 4 presents the circuit design in detail; Sect. 5 shows synthesis results in terms of performance and area on FPGA; finally, Sect. 6 summarizes the main conclusions.

2 Theoretical background

2.1 Dynamic speckle

When a coherent light source (laser) illuminates a non-polished surface, surface roughness causes a random interference phenomenon, known as "speckle". This phenomenon is originated by the different path lengths between scattering points of the surface and an observation source. Optical systems can be used for scanning the phenomenon and registering it in successive intensity images. The images show peculiar speckle patterns depending on surface roughness, the incoming light wavelength, and the numerical aperture of the imaging optical system (see Fig. 1).

The sequences of intensity images are captured at fixed time intervals and the intensity variation of each pixel through the sequence determines a one-dimensional signal called "time history speckle pattern" (THSP) [13]. These signals show a highly random behavior; hence, the sample characterization involves the discovery of hidden patterns in a noisy frame. Braga et al. [3] showed that THSP did not behave as white noise, providing evidence that these signals contain reliable information of the dynamics of the sample.

Most of the descriptors used for monitoring the dynamic speckle activity require a large number of images. Most implementations are based on programs running on general purpose processors, so they generally cannot be carried out in real time. Then special-purpose devices are considered of interest when a real-time process is a must.

2.2 Rough-fuzzy granular descriptor (RFGD)

In speckle image processing, the speckle intensity evolution can be seen as temporal granulation [5], where levels of brightness seen as information granules are merged.

Information granules are collections of entities, grouped together due to their similarity, functional adjacency, indistinguishability, coherence or alikeness [21]. In the signal processing, the information granules contribute to condensing a signal and representing it as a set of temporal granules through an abstraction mechanism that synthesizes the information [2]. The process is determined by the suitable choice of the granules of information.

To generate information granules in FGD, several fuzzy sets are defined into the intensity values domain of the THSP. For a context of intensity values $V = [0, 255]$ and $v_i \in V$, a fuzzy set F_k is defined by a membership function $\mu_{F\_k}(v_i)$ that takes gradual values in the real interval [0,1] (Eq. 1).

In the work presented by Dai Pra et al. [5], three ordered fuzzy sets $k \in \{dark, medium, light\}$ are defined on the one-dimensional space V. These fuzzy sets are represented by partially overlapped trapezoidal membership functions μ_{F_k}, where the membership function values different from zero define the support (supp) of the fuzzy set (Eq. 2).

$$F_k = \{(v_i, \mu_{F_k}(v_i)), v_i \in V \text{ and } \mu_{F_k} \in [0, 1]\} \quad (1)$$

$$\text{supp } F_k = \{v_i \in V, \mu_{F_k}(v_i) > 0\} \quad (2)$$

The parameterization of the membership functions of the fuzzy sets is crucial for a granule count that will depend

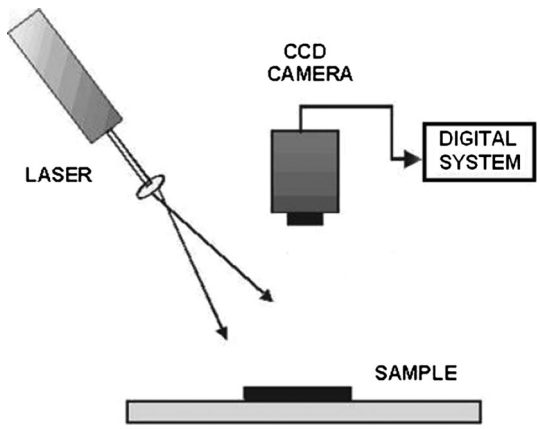

Fig. 1 Optical setup

on the dynamics of the speckle patterns. To overcome the variations of the intensity distributions among the cases of application, we propose a simple and fast adaptive method, which has been successfully tested in previous works [5– 7]. Hence, to establish the parameters of the trapezoidal membership functions, an intensity values histogram of 256 bins is built from a representative sample of the image sequence. The frequency distribution of the histogram is partitioned in five intensity value ranges so that a balanced quantity of elements belongs to each one. Fuzzy sets are defined considering that the first, middle and last intensity ranges are associated with the maximum membership values (core values) of the *dark*, *medium* and *light* fuzzy sets, respectively. The remainder intervals take decreasing and increasing membership values to both sides of the partition. In this case, total overlapping of the contiguous sets is assumed in these intervals (see Fig. 2a, b).

In this work, we adopt the approach of Rough Set theory to define the intensity information granules. Given that the Rough sets are defined from the knowledge acquired through the methodology described above, we propose a Rough-Fuzzy Granulation process [15]. The rough sets philosophy is based on the assumption that a certain amount of information (data and knowledge) is associated with all objects of a universe U, expressed by means of some attributes used to describe the object.

Objects that have the same description are inseparable (similar) with respect to the set B of attributes considered. This relationship of inseparability is the mathematical basis of the theory, and it induces a partition of the universe of objects U into inseparable blocks $[x]_B$. Any subset X of the universe U can be expressed in terms of these blocks exactly or approximately. In this way, the set X can be characterized by two ordinary sets called *lower approximation* $(B - lower : \underline{B}X)$ and *upper approximation* $(B - upper : \overline{B}X)$ [17]:

- The lower approximation of X consists of all blocks that are subsets of X, that is to say that definitely belong to X, $\underline{B}X = \{x \in U : [x]_B \subseteq X\}$.
- The upper approximation consists of all blocks that could belong to X and have a non-empty intersection with X, that is to say that definitely belong to X, and blocks that cannot be determined with certainty belonging to X, $\overline{B}X = \{x \in U : [x]_B \cap X \neq \emptyset\}$.

In order to facilitate the encoding of the intensity range in well-defined regions, given each fuzzy set, the approximations of the corresponding rough set are defined. In this particular case, where the object (pixel) has a single attribute (intensity), it is defined in the universe $V \in [0, 255]$. Considering the parameters of the trapezoidal membership functions of the F_k fuzzy sets, with $k \in \{dark, medium, light\}$, the following rough set approximations are obtained: $\underline{dark}, \overline{dark}, \underline{medium}, \overline{medium}, \underline{light}$, and $\overline{light}$.

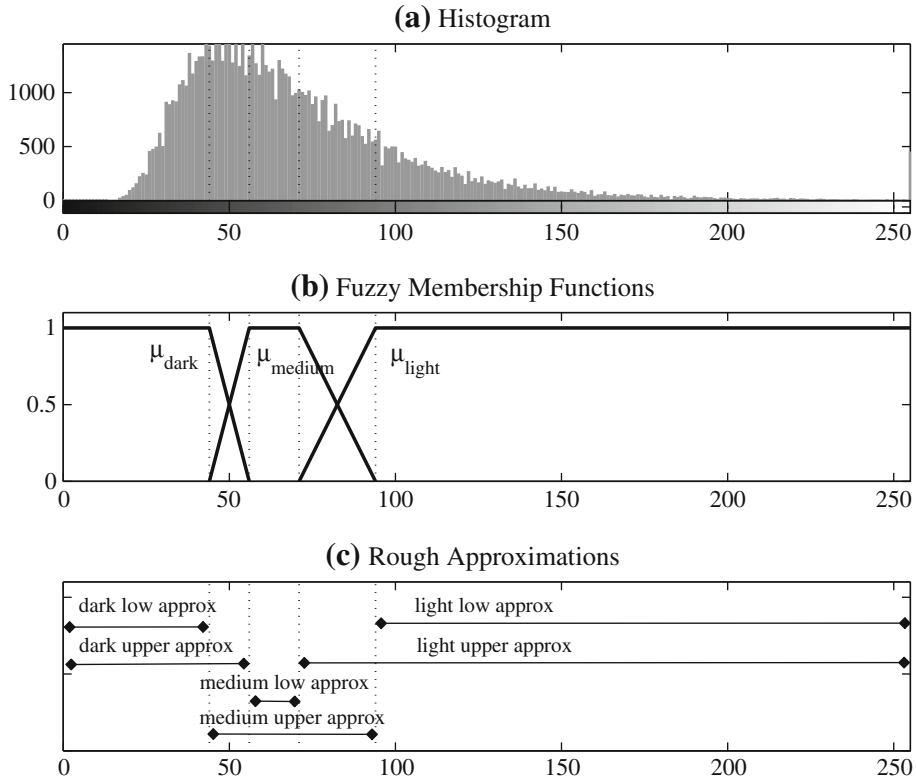

Fig. 2 Intervals definition

- $\underline{dark} = \{v_i \in V : [v]_{dark} \subseteq V, \mu_{dark}(v_i) = 1\}$
- $\overline{dark} = \{v_i \in V : [v]_{dark} \cap V \neq \emptyset, v_i \in \text{supp } \mu_{dark}(v_i)\}$
- $\underline{medium} = \{v_i \in V : [v]_{medium} \subseteq V, \mu_{medium}(v_i) = 1\}$
- $\overline{medium} = \{v_i \in V : [v]_{medium} \cap V \neq \emptyset \; v_i \in \text{supp } \mu_{medium}(v_i)\}$
- $\underline{light} = \{v_i \in V : [v]_{light} \subseteq V, \mu_{light}(v_i) = 1\}$
- $\overline{light} = \{v_i \in V : [v]_{light} \cap V \neq \emptyset, v \in \text{supp } \mu_{light}(v_i)\}$

These approximations arise from the five well-defined intensity regions obtained by partitioning the frequency distribution of the histogram. Table 1 shows the dependence of each of the five regions (clearly shown in Fig. 2c) with the approximations of the rough sets previously computed.

Each THSP is defined as a discrete signal S of elements $s_n \in [0, 255]$. Each granule of the S signal is defined as a sequence of elements belonging to the same rough set upper approximation ($\overline{R}$), whose membership values are now restricted to {0,1}. Figure 3 shows a fragment (51 samples) of speckle signal and its decomposition in granules (*dark*, *medium* and *light*) according to the Rough sets previously defined. As the $\overline{R}_k$ sets are overlapped, such will be the case for the granules.

Table 1 Regions equivalence

Regions	Fuzzy set	Rough set
G_0	$\mu_{dark}(v) = 1$	$\underline{dark}$
G_1	$\mu_{dark}(v) \in (0,1) \wedge \mu_{medium}(v) \in (0,1)$	$\overline{dark} \cap \overline{medium}$
G_2	$\mu_{medium}(v) = 1$	$\underline{medium}$
G_3	$\mu_{medium}(v) \in (0,1) \wedge \mu_{light}(v) \in (0,1)$	$\overline{medium} \cap \overline{light}$
G_4	$\mu_{light}(v) = 1$	$\underline{light}$

The RFGD is computed using Eq. 3 applying Q_T to each THSP or S signal.

$$Q_T(S) = \sum_{k=0}^{Z-1} |suc_{n,k}(\overline{R}_k(s_n))|/T, \quad \text{with } n = 2, 3, \ldots, T. \tag{3}$$

$$suc_{n,k} = \begin{cases} 1 & \text{if } \overline{R}_k(s_{n-1}) = 0 \wedge \overline{R}_k(s_n) = 1 \\ 0 & \text{in other case} \end{cases} \tag{4}$$

where T is the number of instances considered, n corresponds to each instance in T, k identifies each rough set, Z is the quantity of rough sets considered; and |.| indicates cardinality, i.e., the quantity of granules registered in T instances. In real time, to monitor the dynamics of the phenomenon, the Q_T must be calculated when a new pixel is sent from the camera controller, taking into account the quantity of previously accumulated granules. Figure 3 shows the corresponding computed Q_T and the zone coding employed in the algorithm explained in Sect. 3.

3 Activity computation algorithm

Considering the general case of Z rough sets, associated to each fuzzy concept numbered from 0 to $Z - 1$, one defines a coding system using Z bits $x_{Z-1}, x_{Z-2}, \ldots, x_0$; c zone-code will be set to $x_c = 1$, $x_{i \neq c} = 0$ when there are no overlapping zones, while overlapping zone c and $c + 1$ will be set to $x_c = x_{c+1} = 1$, $x_{i \neq c, c+1} = 0$.

Figure 3 shows the zone coding for three rough sets. Let the vector $x_{Z-1}(i), x_{Z-2}(i), \ldots, x_0(i)$ be the coded intensity of a given pixel at instance i. The quantity of granules Q_T of a given pixel signal S may be computed as

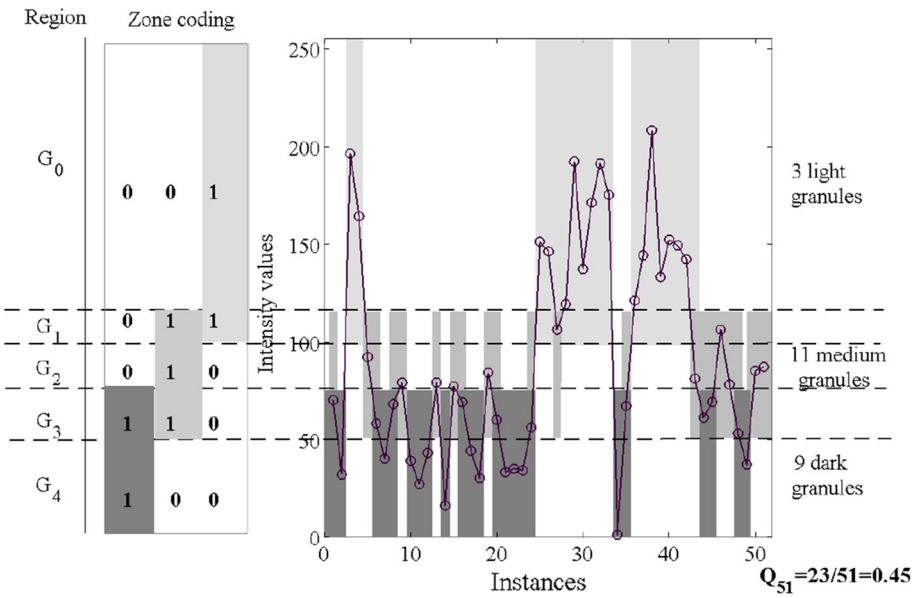

Fig. 3 Granules density computation in an arbitrary signal

$$Q_T(S) = \sum_{i=0}^{T} (\overline{x_{Z-1}(i)} \wedge x_{Z-1}(i+1) + \overline{x_{Z-2}(i)} \wedge x_{Z-2}(i+1) + \cdots + \overline{x_0(i)} \wedge x_0(i+1))/T \quad (5)$$

where $x_c(0) = 0$, $\forall\ c\ in\ [0, Z-1]$ to cope with the first instance. Without ambiguity, $\sum$ and $+$ stand for the arithmetic integer sum, while symbols $\overline{x}$ and $\wedge$ stand for the Boolean complementation and the Boolean AND, respectively.

Clearly, whenever bit $x_c(i)$ changes from 0 to 1 at instance $i+1$, an integer 1 is added to the sum in Eq. 5, as a new granule is detected in the S signal.

The data source (images) consists of N_f frames of N_p pixels with their respective intensity values:

$$N_p = N_r \times N_c \quad (6)$$

N_r and N_c are the respective quantities of rows and columns of each frame.

The activity index of a given pixel p is computed according to Eq. 5.

The following algorithm consists of three steps. The first two steps are actually preset procedures: the first step sets the intensity histogram; the second step computes the actual intensity region limits, featuring a balanced distribution of pixels in each region. The third step computes granule quantities and activity indexes (i.e., descriptors) as Q_T.

3.1 Gray-level histogram setup

For intensity values between 0 and 255, 8 bits are used. The first captured frame is taken as data for this phase, so N_p pixels are involved (see Eq. 6). For counting purposes, 256 bin-registers are defined, one bin for each intensity value. Thus, the gray-level histogram is built up as a 256-word vector h. Word $h(i)$ is formed by the quantity of pixels with intensity i. In short, for every pixel with intensity i in the frame, one unit will be added in the ith bin. The pseudo-code in Algorithm 1 builds up vector h.

Algorithm 1 Histogram computation (result in h) from the first $N_r \times N_c$-pixel image. $pixel_{i,j}$ stands for the intensity value captured from the image.

```
{The gray-level histogram, h, has 256 bins}
for i in 0 to 255 do
  h(i) ← 0
end for
for i in 0 to N_r − 1 do
  for j in 0 to N_c − 1 do
    h(pixel_{i,j}) ← h(pixel_{i,j}) + 1
  end for
end for
```

3.2 Computation of region limits

The next step is the computation of the region limits. The Z rough sets are overlapped, generating Z non-overlapped regions and $Z-1$ regions where the functions are overlapped (Fig. 2), resulting in a total of $(2 \cdot Z - 1)$ regions G.

Two values are necessary to define each region. Those limit values are in [0, 255]. The left limit of the leftmost region is 0 and the right limit of the rightmost region is 255. Now, the region limits are defined in such a way that an (closest to) equal number of pixels would belong to each region. Therefore, the gray-level histogram is used as input for the region limits assignment. For this purpose, a bin look-up procedure is carried out by adding bin stored values from bin 0 on, up to the point when the accumulated sum is closest to

$$N_m = N_p/(2Z-1) \quad (7)$$

Let bin i be the last one to be included in the first counting, i will be the upper limit of the first region, and $i+1$ the lower limit of the next region. This process is then repeated from bin $i+1$ up to the point when the accumulated count is closest to N_m. Figure 2 displays five quite regular regions as the intensity values quantity has been assumed to be equal for all regions. The pseudo-code in Algorithm 2 computes the region limits.

Algorithm 2 Region limits ($region_k$) computation.

```
{Z is the number of overlapping membership functions generating 2 · Z − 1 regions}
{The range of intensity values [0,255] is partitioned in such a way that a similar number of elements could be included in each one of the 2 · Z − 1 regions}
{region_0 = 0; region_{2·Z−1} = 255}
N_m ← N_p/(2 · Z − 1) {N_m: #pixels in each region}
sp ← 0 {auxiliary}
i ← 0
for k in 1 to 2 · Z − 2 do
  while sp < N_m · k do
    sp ← sp + h(i)
    i ← i + 1
  end while
  region_k ← i
end for
```

3.3 Granular computing

At the granular computing stage, all the S signals related to the corresponding N_p pixels are sequentially built up and analyzed to weigh up activities. The pixels are coming from the camera serially, i.e., the first N_c pixels from the first row and so on, up to the N_rth row; thus, an iterative algorithm is proposed. For each incoming pixel (with intensity data), the membership function value is computed and compared with the previous value for the same pixel. The most intuitive method would first determine the related intensity region through comparison of the pixel intensity

code with the respective region limits. Then, by matching up to the pixel data at the preceding instance, up to two new granules can be detected and added to the counter. An alternative rests upon a preliminary implementation to achieve a functional mapping of the intensity code 8-bit vector onto a zone coding Z-bit vector. Further on, the occurrence of a new granule is detected whenever, between two successive instances, the zone coding vector value changes in such a way that some bit switches from 0 to 1.

The previous membership zone coding values for every pixel are stored in a memory-flag of $(N_r \times N_c)$ Z-bit words. Actually, the pseudo-code in Algorithm 3 computes formula 5. For this sake one defines the pseudo-Boolean operator

$$N_{i,j}(k+1) = \overline{x_{Z-1}(k)} \wedge x_{Z-1}(k+1) + \overline{x_{Z-2}(k)} \wedge x_{Z-2}(k+1) + \cdots + \overline{x_0(k)} \wedge x_0(k+1) \quad (8)$$

associated without ambiguity to zone coding vector (i,j) : $x_{Z-1}(k+1), x_{Z-2}(k+1), \ldots, x_0(k+1)$

Algorithm 3 Activity index computation, results in $A_{i,j}(k+1)$.

```
{N_f is the number of N_p-pixel frames ( = 2^b )}
{One granule counter f_{i,j} per pixel}
for i in 0 to N_r, j in 0 to N_c do
   f_{i,j} ← 0
end for
k ← 0
while k < N_f − 1 do {Current frame}
   for i in 0 to N_r − 1 do {Current frame row}
      for j in 0 to N_c − 1 do {Current frame column}
         compute N_{i,j}(k + 1) {granules counting}
         f_{i,j} ← f_{i,j} + N_{i,j}(k + 1)
         A_{i,j} ← f_{i,j}/(k + 1)
      end for
   end for
end while
```

3.4 Memory requirements

Granule counters: $N_p = N_r \times N_c$ granule counters are needed to cope with every pixel activity monitoring. This involves a memory requirement, in bits, of

$$C_{mem} = N_p \times L \quad (9)$$

where L stands for the register-cell size. As the maximum quantity of instances (frames) is $N_f = 2^b$, $b+1$ bits are required for L. Actually, b bits are enough to count from 0 up to $2^b - 1$, but two granules could be involved at each counting step (for the overlapping feature), so an extra bit is needed.

Histogram setup: Assuming intensity values in [0, 255], 256 bin-memory registers are needed to store quantities in [0, N_p]. The memory requirement in bits is given as

$$H_{mem} = 256 \times \lceil \log_2 N_p \rceil \quad (10)$$

Zone-code memory: At each new instance, the pixel intensity (expressed by a zone-code) has to be matched to the zone-coding of the same pixel at the preceding instance. So zone-coding flags have to be stored and this requires the following amount of memory

$$F_{mem} = N_p \times Z. \quad (11)$$

In short, memory requirements are dominated by the counters and the zone-code memories. In order to obtain maximum performance, external memory access should be reduced as much as possible. Table 2 shows memory requirements for different image configurations. As shown in Sect. 5, there are off-the-shelf programmable logic devices with these amounts of on-chip memory.

4 Digital circuit

Figure 4 shows the digital circuit developed for the proposed activity computation algorithm. It consists of three main memories, described in Sect. 3.4, and four main modules. There is one module for each algorithm in Sect. 3, and an additional output module that computes the quotient of the current granule count by the number of processed frames.

The circuit is parameterized by the maximum number of frames, $N_f = 2^b$, the number of rows and columns in the frame, and the activity index precision (i.e., the quotient). Circuit inputs are provided by a camera controller that drives a *start* signal to initiate a new computation; *pxIn*, a grayscale 8-bit signal with the intensity values of a single pixel at a time; *pxInAddr*, a pixel identification in [0, $N_p - 1$] ; and *frIn*, a b-bit signal for the frame number. The outputs are Q, a p-bit signal (p stands for the user-defined precision) for the current pixel activity index; *pxOutAddr*, a pixel identification in [0, $N_p - 1$] related to the activity index in Q; and *frOut*, a b-bit signal for the frame number related to Q.

Table 2 Memory requirements in bits for different image setups

N_p	N_f	Z	C_{mem}	F_{mem}	H_{mem}	Total
512 × 512	128	3	2,097,152	786,432	4,608	2,888,192
512 × 512	64	3	1,835,008	786,432	4,608	2,626,048
400 × 400	128	3	1,280,000	480,000	4,426	1,764,426
400 × 400	64	3	1,120,000	480,000	4,426	1,604,426
350 × 350	128	3	980,000	367,500	4,328	1,351,828
350 × 350	64	3	857,500	367,500	4,328	1,229,328

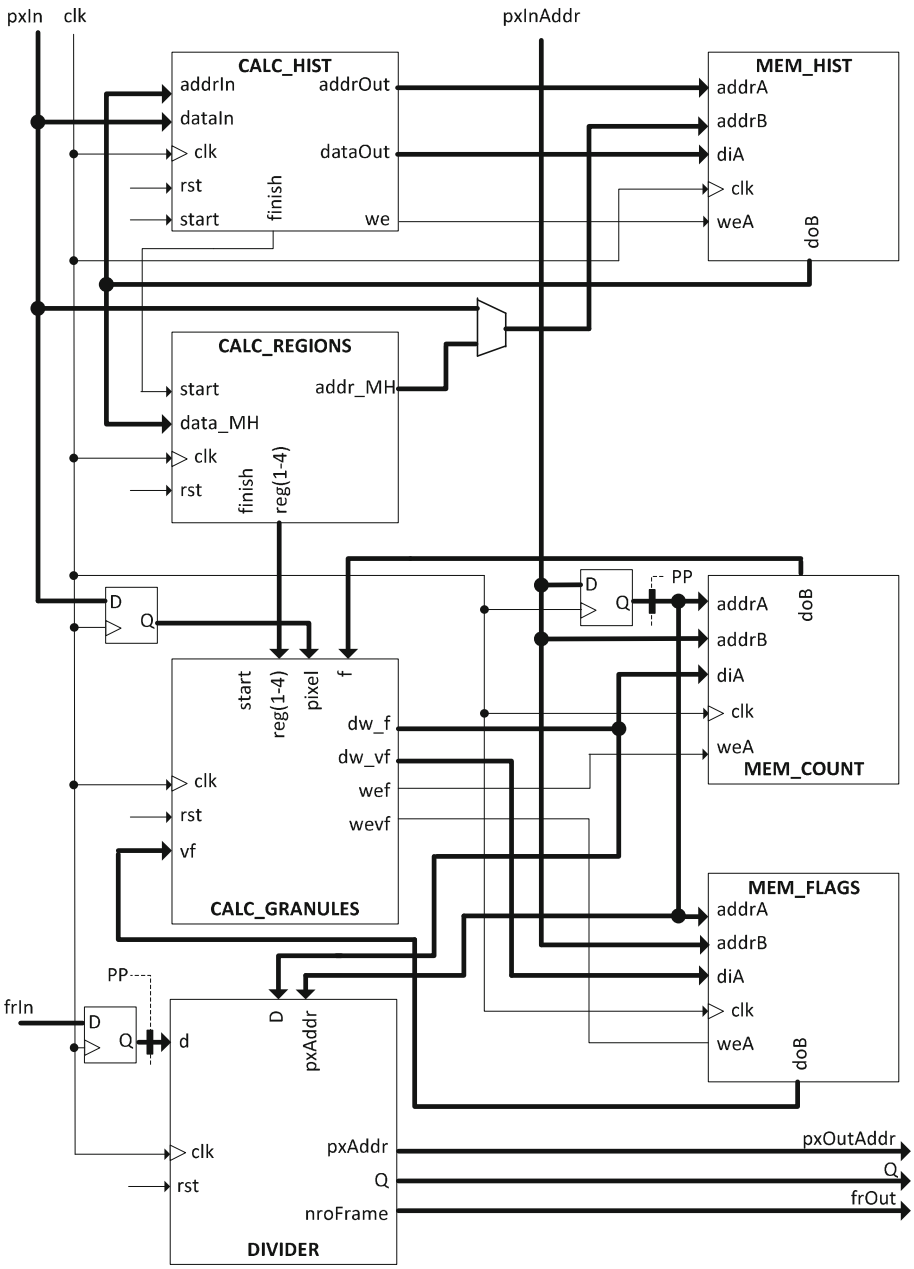

Fig. 4 Digital circuit to compute the activity index of all pixels in each new frame

4.1 The circuit to obtain the histogram

The first frame captured by the camera is analyzed in this module. The 256 bin-registers defined in Sect. 3.1 are stored in the *MEM_HIST* memory. The module reads this memory addressed by the current pixel intensity in [0, 255]. The pixel count at this address is increased by one and written back in the same register at the next clock cycle, while a new pixel is reaching the read stage in a pipelined fashion. To achieve this level of parallelism, *MEM_HIST* must be a dual-port memory.

4.2 The circuit to compute the region limits

This circuit computes the region limits as described in Sect. 3.2, using the histogram stored in *MEM_HIST*. This is achieved by accessing *MEM_HIST* sequentially and accumulating bin-register values in the *AccData* register up to when N_m is exceeded (see Fig. 5). Then the current *HIST_MEM* address is saved in the corresponding *Region* register. This process is repeated four times as $2 \cdot Z - 2$ *Region* registers are needed considering three rough-fuzzy zones in this circuit.

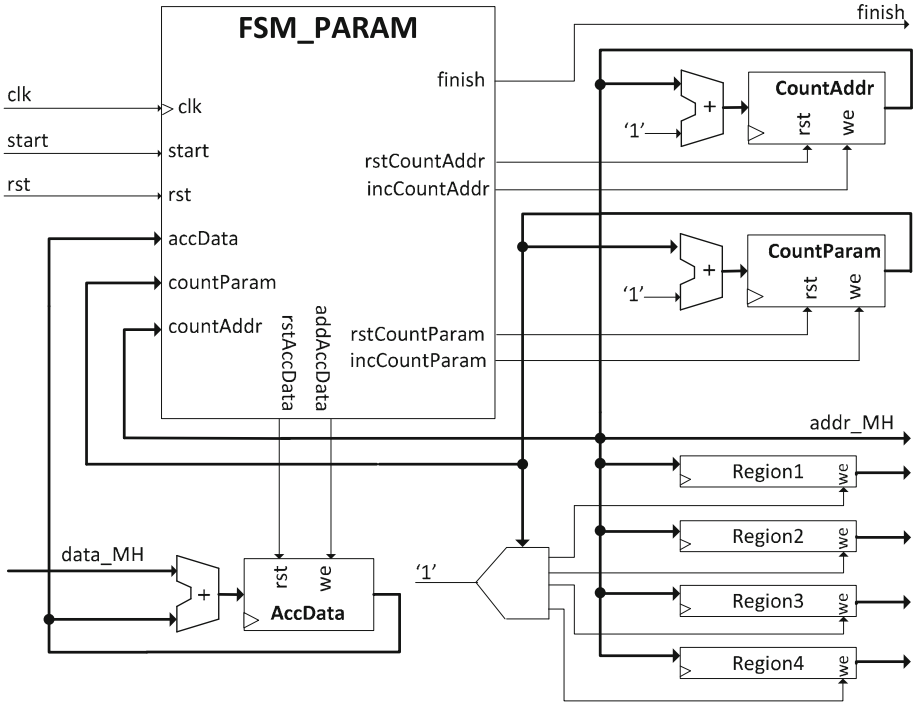

Fig. 5 Digital circuit to compute the region limits

4.3 The circuit to count granules

This circuit updates the granule count for each image pixel in a sequence of frames as explained in Sect. 3.3. However, in order to obtain better throughput, the time-consuming division operation is implemented in a separate division module in a pipelined fashion, as described in Sect. 4.4.

A new pixel from the current image is processed at each clock cycle. The zone coding of the same pixel in the previous image is read from *MEM_FLAGS* memory and compared with its current zone coding (see Fig. 6). The zone coding of the current pixel is obtained by comparing its intensity with the previously computed region limits. If the pixel intensity changes to new zones, up to two, the corresponding granule count stored in *MEM_COUNT* is increased accordingly by 0, 1, or 2 in the next clock cycle. *MEM_FLAGS* is also updated with the current zone coding in the next clock cycle.

In order to reach the maximum speed, *MEM_COUNT* and *MEM_FLAGS* are read and written on a single clock cycle. Therefore, dual-port memories are selected.

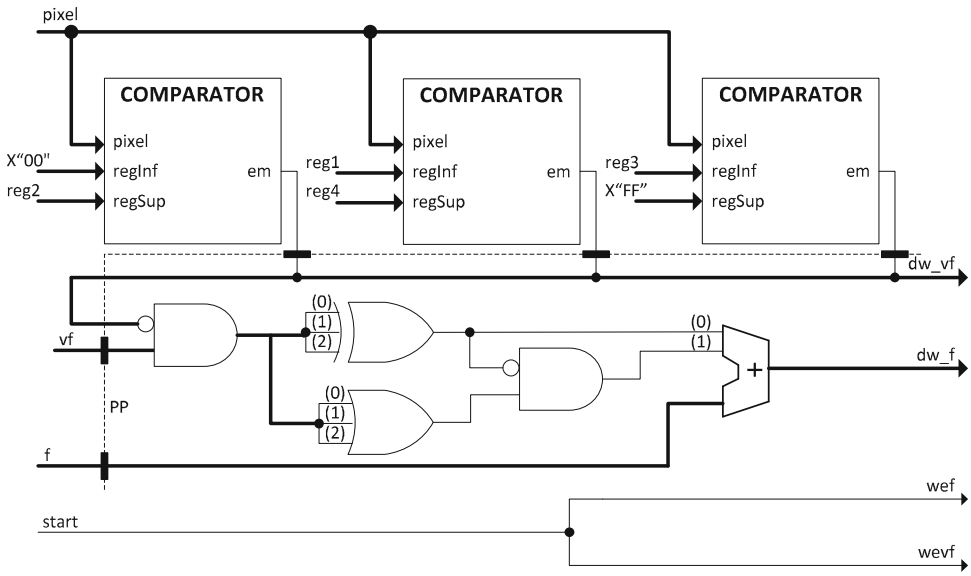

Fig. 6 Digital circuit to compute granule counts

4.4 Division for activity computation

The classic fixed-point restoring division algorithm is used and customized to compute the activity index for each pixel at every new frame [8]. That is $f_{i,j}$ / $(k+1)$ of Algorithm 3.

Granule counters store $f_{i,j}$ as $(\lceil \log_2 N_f \rceil + 1)$-bit integer numbers, the same operand width as for the current frame number $k+1$. As a new pixel is received from the camera at every clock cycle, a pipelined divider circuit has been designed in order to start (and complete) the computation of a new quotient at every new clock cycle. As no normalization is applied to the operands, some leading zeros could be obtained for the quotients.

Section 3 shows that granule counters can be increased by at most two units per frame. Thus, the activity index is in [0, 2]. With p bits of fractional precision, an XX.X $\cdots$ X format can be used. Moreover, as the maximum possible quotient is 2 (i.e., 10.0 $\cdots$ 0), almost half of the combinations could be wasted. In consequence, the division module generates, with a latency of $p+2$ clock cycles, an exact quotient of the form X.X $\cdots$ X for results <2, and the approximation 1.1 $\cdots$ 1 when the quotient is exactly 2. Once the histogram and region limits are computed, the module that counts granules has a latency of 2 clock cycles. Then the total latency of the circuit computing the activity indexes is $p+4$ clock cycles.

5 Experiments

Table 3 shows the synthesis results carried out on FPGA based on a Xilinx Virtex-6 device (xc6vlx130t ff484 −3). The first column displays image sizes. The minimum clock period, t_p, the number of flip-flops, the number of slice-LUTs, and the number of RAM blocks (RAMB36E1 + RAMB18E1) are shown for THSPs of 64 and 128 instances.

Although the area consumed in terms of programmable resources (#ff and #LUT) is negligible in a modern FPGA device, the area consumption grows slightly as the frame size and the number of analyzed frames increase. However, the size of the required embedded memory grows linearly with the number of analyzed pixels.

Regarding the minimum clock period, the increase in t_p is larger than the one in area for the consumed programmable resources. This leads to the following critical path study. In FPGA, when a single memory needs a significant number of embedded block RAM resources to be implemented, the delay in reading and writing data increases accordingly. The minimum clock period increase in the granules counting module occurs because it is necessary to read big memory structures like *MEM_COUNT* and *MEM_FLAGS*, then process the information (comparators, switch zone logic, and adder), and finally write results back. Thus, the critical path can be improved by splitting this particular pipeline stage. The minimum clock period is now $t_{p,opt}$ in Table 3. The small black rectangles in the equitemporal dotted line *PP* in Figs. 4 and 6 stand for the additional registers needed to implement this timing optimization. Critical path analysis enables a trade-off between clock period and latency. For example, the number of division stages can be reduced to half without penalty in the original critical path because it is located in the non-optimized granules counter module.

Table 3 Implementation results on FPGA (Virtex-6 Family). The precision of the activity index is 8 bits

#inst	64	128
128 × 128		
t_p (*ns*)	4.02	4.04
$t_{p,opt}$ (*ns*)	3.48	3.48
#ff	311	331
#LUT	377	398
#RAMB	4 + 3	5 + 2
256 × 256		
t_p (*ns*)	4.86	4.87
$t_{p,opt}$ (*ns*)	3.52	3.52
#ff	319	334
#LUT	397	416
#RAMB	20 + 1	22 + 1
512 × 512		
t_p (*ns*)	5.42	5.43
$t_{p,opt}$ (*ns*)	3.52	3.52
#ff	329	349
#LUT	404	423
#RAMB	80 + 1	88 + 1

Considering a latency of 12 clock cycles (+1 if the optimization is performed) for 8-bit precision activity indexes, it is clear that this implementation can be used for very challenging real-time applications. In the optimized Virtex-6 implementation a throughput of about 1,000 fps can be obtained for 512 × 512 frames using a single device.

Table 4 shows first-order estimations of power and energy consumption obtained by using Xilinx Xpower tool. This analysis only includes the consumption of the FPGA core at 1V. The external load and the I/O consumption are not considered. The circuit frequency is 250 MHz (i.e., a period of 4 ns) for the estimation. As the proposed circuit is rather small and hence the consumed area in the FPGA is small, almost all the power and energy consumption is due to leakage current. In fact these values are much lower than the ones obtained using general purpose computers.

Table 4 Power and energy estimations on FPGA (Virtex-6 Family) for the optimized circuit

#inst	64	128
128 × 128		
Power (W)	1.129	1.131
Energy per pixel (nJ)	4.516	4.524
256 × 256		
Power (W)	1.085	1.089
Energy per pixel (nJ)	4.340	4.356
512 × 512		
Power (W)	1.113	1.117
Energy per pixel (nJ)	4.452	4.468

6 Conclusions

The speckle activity descriptor FGD, whose efficacy has been previously reported to identify both stationary and non-stationary dynamics, was reformulated as the RFGD to improve its computational performance in a digital circuit implementation. The RFGD descriptor is presented at different levels of abstraction, from a mathematical description in Sect. 2, to an algorithm expressed as pseudo-code in Sect. 3, and finally the corresponding digital circuit as a portable HDL description in Sect. 4.

The designed circuit is developed by a portable description in a hardware description language (HDL). While the circuit is implemented on a specific FPGA technology, it can be synthesized on FPGAs of other vendors and VLSI.

Since the proposed algorithm is of low computational cost, the corresponding digital circuit is designed using a small and pipelined circuitry, consisting of 1 adder, 6 comparators, and an additional logic level for the switch zone logic in the granules computation module, and 9 subtractors and 9 multiplexers for the division module. Therefore, the circuit can perform high-speed granular computing of N_p THSP signals provided by a camera in real time. Although the data path is small and pipelined, the circuit needs access to a memory whose size depends on image size and the number of processed frames. Thus, best results are obtained if this memory can be implemented on-chip, which is the case in the presented experimental results. The modules to obtain the histogram and to compute the region limits are less important in the timing analysis because they are involved only in the preset stage.

Acknowledgments This work was partially supported by FASTA University, UNCPBA University, and National University of Mar del Plata investigation projects fund, and by the Agencia Nacional de Promoción Científica y Tecnológica, Argentina, through projects PICT-2008-1430 and PICT-2009-0041.

References

1. Arizaga, R., Cap, N.L., Rabal, H., Trivi, M.: Display of local activity using dynamical speckle patterns. Opt. Eng. **41**(2), 287–294 (2002). doi:10.1117/1.1428739
2. Bargiela, A., Pedrycz, W.: Granular Computing. An Introduction. The Springer International Series in Engineering and Computer Science, vol. 717. Springer, Berlin (2002)
3. Braga, R., Silva, W., Sáfadi, T., Nobre, C.: Time history speckle pattern under statistical view. Opt. Commun. **281**(9), 2443–2448 (2008). doi:10.1016/j.optcom.2007.12.069
4. Braga, R.A., Fabbro, I.M.D., Borem, F.M., Rabelo, G., Arizaga, R., Rabal, H.J., Trivi, M.: Assessment of seed viability by laser speckle techniques. Biosyst. Eng. **86**(3), 287–294 (2003). doi:10.1016/j.biosystemseng.2003.08.005
5. Dai Pra, A.L., Passoni, L.I., Rabal, H.: Evaluation of laser dynamic speckle signals applying granular computing. Signal Process. **89**(3), 266–274 (2009). doi:10.1016/j.sigpro.2008.08.012
6. Dai Pra AL, Passoni LI, Rabal H (2009b) Temporal granular computing in dynamic speckle signal processing. In: XIII Reunión de Trabajo en Procesamiento de la Información y Control, Rosario, Argentina
7. Dai Pra, A.L., Passoni, L.I., Rabal, H.J.: Fuzzy granular computing and dynamic speckle interferometry for the identification of different thickness of wet coatings. INFOCOMP J. Comput. Sci. **8**(4), 45–51 (2009)
8. Deschamps, J.P., Bioul, G., Sutter, G.: Synthesis of Arithmetic Circuits: FPGA, ASIC and Embedded Systems. Wiley, New York (2006)
9. Fricke-Begemann, T., Gülker, G., Hinsch, K.D., Wolff, K.: Corrosion monitoring with speckle correlation. Appl. Opt. **38**(28), 5948–5955 (1999). doi:10.1364/AO.38.005948
10. Fujii, H.: Apparatus for monitoring bloodstream. US Patent 4862894 (1989)
11. Fujii, H., Asakura, T., Nohira, K., Shintomi, Y., Ohura, T.: Blood flow observed by time-varying laser speckle. Opt. Lett. **10**(3), 104–106 (1985). doi:10.1364/OL.10.000104
12. Mavilio, A., Fernndez, M., Trivi, M., Rabal, H., Arizaga, R.: Characterization of a paint drying process through granulometric analysis of speckle dynamic patterns. Signal Process. **90**(5), 1623–1630 (2010). doi:10.1016/j.sigpro.2009.11.010. Special Section on Statistical Signal Array Processing
13. Oulamara, A., Tribillon, G., Duvernoy, J.: Biological activity measurement on botanical specimen surfaces using a temporal decorrelation effect of laser speckle. J. Mod. Opt. **36**(2), 165–179 (1989). doi:10.1080/09500348914550221
14. Pajuelo, M., Baldwin, G., Rabal, H., Cap, N., Arizaga, R., Trivi, M.: Bio-speckle assessment of bruising in fruits. Opt. Lasers Eng. **40**, 13–24 (2003). doi:10.1016/S0143-8166(02)00063-5
15. Pal, S.K., Mitra, P.: Case generation using rough sets with fuzzy representation. IEEE Trans. Knowl. Data Eng. **16**(3), 292–300 (2004)
16. Passoni, I., Dai Pra, A.L., Rabal, H., Trivi, M., Arizaga, R.: Dynamic speckle processing using wavelets based entropy. Opt. Commun. **246**(1–3), 219–228 (2005). doi:10.1016/j.optcom.2004.10.054
17. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Inf. Sci. **177**(1), 3–27 (2007)
18. Rabal HJ, Braga RA (eds) (2008) Dynamic Laser Speckle and Applications. CRC Press, Boca Raton
19. Sendra, G.H., Arizaga, R., Rabal, H., Trivi, M.: Decomposition of biospeckle images in temporary spectral bands. Opt. Lett. **30**(13), 1641–1643 (2005). doi:10.1364/OL.30.001641
20. Sendra, H., Murialdo, S., Passoni, L.: Dynamic laser speckle to detect motile bacterial response of *Pseudomonas aeruginosa*. J. Phys. Conf. Ser. **90**(1), 012,064 (2007)

21. Zadeh, L.A.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets Syst. **90**(2), 111–127 (1997). doi:10.1016/S0165-0114(97)00077-8

Author Biographies

Elías Todorovich received a computer engineering degree from the State University UNCPBA of Tandil (Buenos Aires), Argentina, in 1997, and a PhD degree in computer engineering from the Universidad Autónoma de Madrid (UAM), Spain, in 2006. From 2003 to 2009 he was an assistant professor in the Department of Computer Engineering, UAM, and is currently a professor at the University FASTA of Mar del Plata, Argentina, and adjunct professor at the UNCPBA. His research interests include programmable logic, digital image processing, and functional verification.

Ana Lucia Dai Pra received the Scientific Computer degree in the National University of Buenos Aires (UBA) in 1975. She has also taken the MSc degree course in Software Engineering in National University of La Plata (UNLP). She is currently a College Professor in the Applied Computing Area at Department of Mathematics of the Engineering School at the UNMDP. Currently, she is co-leader in an R&D project in the fields of Computational Intelligence (CI) and she integrates the research group in Artificial Intelligence Applied to Engineering at the UNMDP. Her research interests are in the areas of Artificial Intelligence, Data modeling and CI techniques, particularly in Fuzzy Systems.

Lucia Isabel Passoni received a degree in electrical engineering with electronic orientation and the PhD degree, both from the National University of Mar del Plata (UNMDP) in 1981 and 2005, respectively. She has also received the MSc degree in University Management from the Social and Economic Science School (UNMDP) in 2001. She is currently a Full Associate Professor in the Circuits Area at the Department of Electronic of the Engineering School at the UNMDP. Currently, she leads an R&D group in the field of Computational Intelligence (CI) within the Bioengineering Laboratory at the UNMDP. Her research interests are in the area of decision support system, including data modeling, signal and image processing with CI techniques: Neural Networks, Fuzzy Systems, Evolutionary computation.

Martín Vázquez received a computer engineering degree from the University UNCPBA of Tandil (Buenos Aires), Argentina, in 2002. He also received the MSc degree in Systems Engineering from UNCPBA, in 2011. He is currently working toward the PhD degree at the UNCPBA. He is an adjunct professor at both the University FASTA of Mar del Plata, Argentina, and the UNCPBA. His research interests include FPGA design, digital arithmetic and embedded systems.

Ezequiel Cozzolino received a computer engineering degree from the University FASTA of Mar del Plata, Argentina, in 2007. Actually he is working at the Remote Sensing laboratory of INIDEP (Fisheries Research and Development Institute). He integrates the research group in Real Time Image Processing in FPGA at the University FASTA. His research interests include programmable logic, digital image processing,embedded systems,satellite image processing, and automation and control systems.

Fernando Ferrara received a computer engineering degree from the University FASTA of Mar del Plata, Argentina, in 2011. He is a teaching assistant at the University FASTA. He is CEO and founder of BohemianLabs. His research interests include digital image processing, programmable logic, and embedded systems.

Gery Bioul received MSc degree in physical aerospace engineering from the University of Liège, Belgium. He worked in digital systems design with Philips Belgium and in computer-aided industrial logistics with several Fortune-100 US companies in the United States and Africa. He has been a professor of computer architecture in several universities mainly in Africa and South America. He is currently professor at the University FASTA of Mar del Plata, Argentina, professor consultant at CAECE University, Mar del Plata, and professor at the State University UNCPBA of Tandil, Argentina. His research interests include logic design and computer arithmetic algorithms and implementations. He is the author of about 70 international papers and patents on fast arithmetic units.