Speckle Signal Processing through FPGA

E. Todorovich(1, 2), M. Vazquez(1, 2), E. Cozzolino(1),
F. Ferrara(1), G. Bioul(1, 2)
1. Universidad FASTA, Mar del Plata, 3145, Calle Gascón,
2. UNCPBA, 399, Calle Pinto, Tandil, Argentina
gbioul@ufasta.edu.ar

A.L. Dai Para(3), L.I. Passoni(1, 3)
3. UNMDP, 4302, J.B. Justo, Mar del Plata, Argentina

Abstract - **This paper introduces Field Programmable Gate Array (FPGA) technology as an alternative platform to implement algorithms for speckle patterns analysis in real time. Functions and algorithmic procedures have been expressed in pseudo languages then in Hardware Description Languages (HDL). For all cases, time performances are presented for the Xilinx Virtex-6 family. Comparisons are also made with PC platform implementations presented in the literature.**

Keywords - speckle patterns; granular computing; real time synthesis; FPGA.

I. INTRODUCTION

Physical surface variations with time can be monitored and controlled through image capture and analysis. The application of Dynamic Speckle Interferometry (DSI) to the drying of paints has been reported in [1-6], while a practical development has been presented as an Adaptive Speckle Imaging for monitoring the formation process of film on a regular surface topography [7]. The purpose of this paper is to customize and implement granular computing algorithms using FPGA technology; this includes procedures for image processing and analysis through real time monitoring. Taking advantage of the increasing performances of modern FPGA devices, some designs are presented with promising perspectives with respect to time and hardware costs. Applications examples are inspired from the algorithms presented in [6]. The basis of the Granular Computing with Fuzzy Sets (GCFS) and the Speckle Signal Processing (SSP) techniques are reviewed in section 2. Section 3 is dedicated to the implementations and hardware requirements on FPGA. Section 4 presents the results in performances, compared to implementations on PC's platform.

II. THEORETICAL BACKGROUND

A. Dynamic Laser Speckle pattern

Interference phenomena take place whenever a coherent light source (laser) illuminates a rough surface. The surface of most materials are extremely rugged in the scale of an optical wavelength ($\lambda \sim 5 \times 10^{-7}$ m.). When nearly monochromatic laser light is reflected in this kind of surface, optical waves which contain several components are originated from the reflection of different microscopic elements of the surface. The interference of coherent waves provides a granular pattern of intensity that is called *speckle*. The phenomenon is originated by the different path lengths between the different scattering points from the surface and the observation source. Optical systems can be used for obtaining a scan of the phenomenon and register it in successive images. The images show peculiar speckle patterns depending on surface roughness, on the incoming light wavelength and on the numerical aperture of the imaging optical system (Fig. 1).

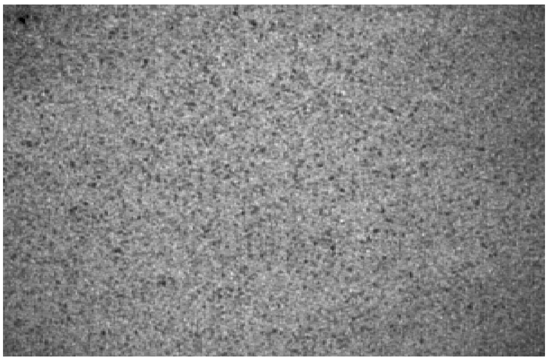
The speckle pattern varies when the illuminated surface presents some type of activity. If the movement is slow the speckles may be recognized within the successive images, but in the presence of higher activity the speckles intensity varies randomly with a rough boiling aspect. The activity variations permit the assessment of diverse phenomena such as microorganism motility [12], seeds viability [13], fruits bruising [14], drying of coatings [1], [3], [4] some of them with considerable economic and/or biological interest.

Images are captured periodically in order to scan the phenomenon. These images capture the intensity of the speckles although they do not permit to identify the illuminated object or surface.

Image activity is associated with pixel intensity variations, within a sequence of images, in a given lap of time. Activity is assumed when there are significant intensity changes in the signals. The notion of significant intensity variation is difficult to define, it is a notion depending of the application at hand, and it can be subjective and uncertain.

The intensity variation of each pixel, through the successive images, determines a one-dimensional signal known as *Time History Speckle Pattern* (THSP) [15]; using this signal the *activity index* is computed.

Figure 1. Raw speckle image



Many efforts have been done to evaluate these activity variations, so many algorithms with different approaches have been proposed [16]. Among them, those algorithms that process the intensity time series belonging to the pixels of an image sequence within the time domain are considered more adequate to be hardware implemented using FPGA. This paper proposes the use of the Granular Computing with Fuzzy Sets (GCFS) methodology [6] to compute the dynamic speckle activity index.

B. *Fuzzy Granular computing*

Granular computing is a technique based on the representation of the information in the form of a number of entities or *information granules* [17].

Granules can be viewed as linked collections of objects (data points, in particular) and drawn together by the criteria of indistinguishability, similarity, proximity or functionality [18]. Granules and the ensuing process of information granulation are vehicles of abstraction leading to the emergence of high-level concepts that support and ease our perception of the surrounding physical and virtual world [17] [19]. For example, the images perceived by human beings are full of information granules, defined by colors, shapes, combinations of colors and shapes. The image processing is naturally split into two main and overlapping levels of processing. At the lower end, one deals with image segmentation, edge detection, noise removal, and so on. At the higher end, the interest rests upon the image description and interpretation, when the level of abstraction depends on the task at hand. This is a spatial granulation.

Another example is the temporal granulation, associated to the processing of times series or signals, where the granulation information is built up over time forming information granules over predefined time intervals. The lower signal processing level deals with measurement and classification of signals, as well as with reduction of interferences and noise. At the higher level, the description, comparison and interpretation of signals is somewhat more complicated because it needs more specific knowledge about the application context.

Granular computing starts with building intelligent systems that are capable of understanding and describing concepts inherently associated to the human activity, to provide a better understanding of the problem at hand, and coming up with an efficient problem-solving strategy. Like a classic strategy, this paradigm breaks a problem down into sub-problems more affordable.

The fuzzy set approach holds useful features to support granular computing and the processes of information granules. Fuzzy sets support modeling of concepts that exhibit continuous boundaries such as tall, old, big, light, which are vague concepts and dependent on the context. The allowed overlapping between fuzzy sets, key feature of the fuzzy theory, allows avoiding the brittleness effect which may occur when swapping one concept for another. This becomes particularly essential when dealing with noise sensitive data.

In the classical set theory, let X be a set of objects, whose elements are denoted x_i, and let A be a subset of X, the membership of each x_i to A is viewed as a function $\mu_A(x_i)$ from X onto {0,1} such that:

$$\mu_A(x_i) = 1 \text{ iff } x_i \in A,\ \mu_A(x_i) = 0 \text{ otherwise} \qquad (1)$$

If the μ_A takes values in the real interval [0, 1], A is called a fuzzy set, while μ_A is the grade of membership of x_i in A. A has no sharp boundaries, as shown at Fig. 2, where five intensity related membership functions are defined. Algebraically, A may be expressed as

$$A = \{(x_i, \mu_A(x_i)), x_i \in X \text{ and } \mu_A(x_i) \in [0,1]\} \qquad (2)$$

One defines the *support* of a fuzzy set A as the subset of X such as

$$Supp\ A = \{x_i \in X, \mu_A(x_i) > 0\} \qquad (3)$$

The α-cut of a fuzzy set A is defined as

$$A_\alpha = \{(x_i, \mu_A(x_i)), x_i \in X \text{ and } \mu_A(x_i) \geq \alpha\} \qquad (4)$$

A_α can be used to reduce the overlapping zones.

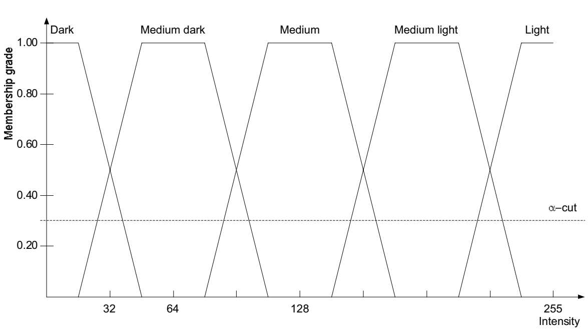
C. *Speckle signal processing procedure concept*

Most of the descriptors used for the activity monitoring [16] using dynamic speckle, require a large amount of images, sometimes involving pre-processing before being processed at regular intervals. Most implementations are based on programs running on general purpose processors; so, they generally cannot be carried out in real time when significant activity, involving a great number of images, is dealt with. Special purpose devices are then of interest when real time process is a must.

As seen in section 2.A the dynamic speckle *activity index AI* characterizes the dynamics of the illuminated sample. It is computed, from the quantity of granules Q_{Nf} generated along each THSP, in Nf instances of intensity observations, as:

$$AI = Q_{Nf}/Nf \qquad (5)$$

Figure 2. Fuzzy sets and membership functions



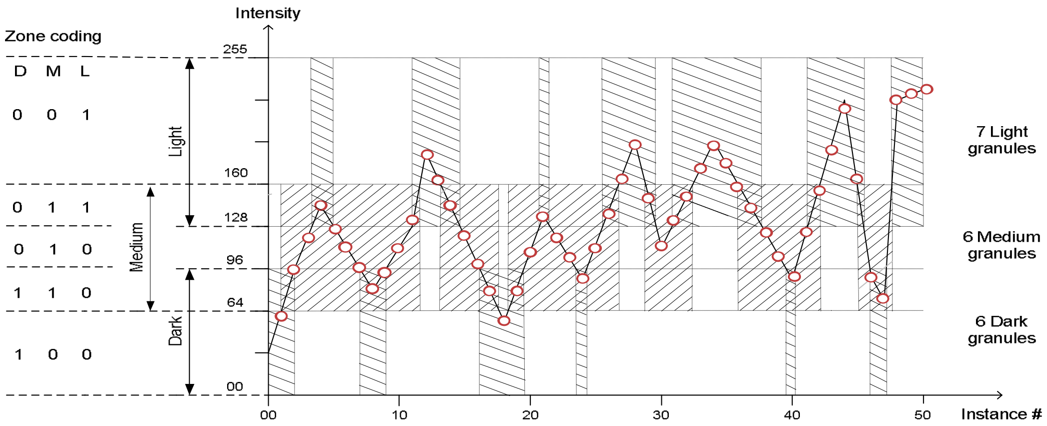


Figure 3. Typical THSP granules diagram

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

In the applications at hand only gray-scale images are considered. Each pixel of the pattern is given an intensity value between 0 (black) and 255 (white). Images from the camera are coded and transmitted through registers associated to the matrix of pixel addresses with correlated intensities. The signals, generated by the pixel intensity changes through the sequence of images, are processed with the finality of identifying underlying activity. This can be modeled with the use of fuzzy granular computing (see section 2.B).

This activity index concept has shown good performances to identify both stationary and non-stationary dynamics, in a wide statistical sense. On a first approach, fuzzy membership functions are defined for three fuzzy sets *dark*, *medium*, and *light*; the selected intensity parameters of the fuzzy functions are obtained through the observation of the gray-level histogram of the first image in the image sequence. Then trapezoidal functions with media overlapping are adopted. In order to detect image contrast effects, membership functions are defined in such a way that equal numbers of elements would belong to each fuzzy set. To control the activity transformations in real time, the activity index must be re-calculated at every instant when a new image is captured, taking into account the number of accumulated granules and the current membership functions values.

For a given pixel, the THSP may be graphically represented as a set of instances with the associated pixel intensity as registered by the camera, this defines the *X* signal. Each granules of the *X* signal is defined as a continuous time sequence of elements belonging to the same fuzzy set. As fuzzy sets are overlapping, such will be the case for the granules. Fig. 3 shows a typical THSP with 3 overlapping main zones tagged *dark*, *medium* and *light*, generating two overlapping zones tagged *medium-dark* and *medium-light*. The example of Fig. 3 illustrates 50 instances creating 19 granules.

D. Computing activity indexes

Considering the general case of R main intensity zones, 0 to R-1, one defines a coding system using R bits x_{R-1}, x_{R-2}, … , x_0; zone k code will be set to $x_k = 1$, $x_{i\neq k} = 0$, while overlapping zone $k \cap k$+1 will be assigned code $x_k = x_{k+1} = 1$, $x_{i\neq k,k+1} = 0$. Fig. 3 shows the coding for three main zones. The *activity index AI* is readily computed as the ratio of the quantity of granules Q_{Nf} and the quantity of instances Nf (5). Let the vector $x_{R-1}(i)$ $x_{R-2}(i)$ … $x_0(i)$ be the coded intensity of a given pixel at instance i, and let Nf be the quantity of instances in the THSP of a given pixel p. The quantity of granules Q_{Nf} may be computed as

$$Q_{Nf}(k) = \Sigma_{i=0\rightarrow Nf}\ (x'_{R-1}(i)\wedge x_{R-1}(i+1) + x'_{R-2}(i)\wedge x_{R-2}(i+1) + \ldots + x'_0(i)\wedge x_0(i+1)), \qquad (6)$$

where $x_{R-1}(0)$ $x_{R-2}(0)$ … $x_0(0)$ is set to (0 0 … 0) to cope with the first THSP granule(s). Without ambiguity, Σ and + stand for the arithmetic integer sum, while symbols ’ and ∧ stand for the Boolean complementation and the Boolean AND respectively.

Clearly, whenever bit $x_j(i)$ changes from 0 to 1 at instance i+1, an integer 1 is added to the sum in (6), as a new granule is detected in the THSP.

The data sources (images) consist of Nf frames of $N_p = N_r$ x N_c pixels with their respective intensity values. N_r and N_c are the respective quantities of rows and columns of each frame.

The activity index of a given pixel k is computed according to (6). The average activity index AI_a will be computed as

$$AI_a = \Sigma_{k=1\rightarrow Np}\ Q_{Nf}(k)/(Nf \times N_p) \qquad (7)$$

III. Activity Computation Algorithm and Memory Requirements

The following algorithm consists of 3 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 granules quantities and activity indexes.

A. Gray-level histogram set-up.

Intensity values are between 0 and 255, 8 bits are used. The first captured frame is taken as data for this phase, so $N_p = N_r$ x N_c pixels are involved. For counting purposes, 256 bin-registers are defined, one bin for each intensity value. The *gray-level histogram* is built up as a 256-word vector *h*. Word *h*(*i*) is formed by the amount of pixels with intensity *i*. In short, for every pixel with intensity *i* in the frame, one unit will be added in the i^{th} bin. The following pseudo-language program builds up vector *h*

```
{Histogram computation (result in h) from the
first (Nr*Nc)-pixel image}
{The gray-level histogram, h, has 256 bins}
h(i)=0 for all i in [0,255]
for i in 0 to Nr-1 do
  for j in o to Nc-1 do
    h (pixel$_{i,j}$) := h (pixel$_{i,j}$) + 1
  end for
end for
```

where $pixel_{i,j}$ stands for the intensity value captured from the image.

B. Region limits computation

The next step is the computation of the membership function parameters. The *R* functions are overlapping trapezoids generating 2*R*-1 regions or zones (Fig. 2), then two values are necessary to define the region limits; this holds for each function but for the rightmost and leftmost ones which need only one. 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 regions 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 adding bin stored values from bin 0 on, up to the point when the accumulated sum is the closest to

$$N_m = N_p / (2R\text{-}1) \tag{8}$$

Let bin *k* be the last one to be included in the first counting, *k* will be the upper limit of the first region and *k*+1 the lower limit of the next region. This process is then repeated from bin k+1 up to the point when the accumulated count would be the closest to N_m. Fig. 2 displays 9 regions (*R*=5) quite regular as the intensity ranges have been assumed to be equal for all regions. In the case of setting the ranges according to the pixel distribution, the geometry would most often be unbalanced as the intensity intervals (defined as the supports of fuzzy sets) would be no more standard. The following pseudo-language program computes the region limits.

```
{Region limits (regionk) computation}
{R is the number of overlapping membership
functions generating 2*R-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*R-1 regions}
{region0 := 0; region 2*R-1 := 255}
Nm := Np/(2*R-1) {Nm: #pixels_in_each_region}
sp := 0; i := 0 {auxiliary}
for k in 1 to 2*R-2 do
  while (sp < Nm*k) do
    sp := sp + h(i)
    i := i +1
  end while
  regionk := i
end for
```

C. Granular computing.

At the granular computing stage all the N_p THSP signals are sequentially built up and analyzed to weigh up activities. The pixels are coming from the camera in a serial way, i.e. the first N_c pixels from the first row and so on, up to the N_r^{th} row; thus an iterative algorithm is proposed. For each incoming pixel (with intensity data), the membership function value is computed and compared to the previous value for this same pixel. The most intuitive method would first determine the related intensity region through comparison of the pixel intensity code with the respective regions limits. Then, 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 *R*-bit vector as defined at section 2.D. Further on, the occurrence of an eventual new granule is detected whenever, between two successive instances, the zone coding vector value changes in such a way that some bit 0 switches to 1. Due to the overlapping feature of the membership functions, up to two bits may switch at the same time, in this case two granules are added to the corresponding granule-counter (see sec. 2.D). The previous membership zone coding values for every pixel are stored in a memory-flag of (N_r x N_c) *R*-bit words. Actually, the following program computes formula (6). For this sake one defines the pseudo-Boolean operator

$$G_{i,j}(k+1) = (x'_{R\text{-}1}(k)\wedge x_{R\text{-}1}(k+1) + x'_{R\text{-}2}(k)\wedge x_{R\text{-}2}(k+1) + \ldots + x'_0(k)\wedge x_0(k+1)), \tag{9}$$

associated without ambiguity to zone coding vector (*i,j*): $x_{R\text{-}1}(k+1)$, $x_{R\text{-}2}(k+1)$, … $x_0(k+1)$.

```
{Activity index computation, results in Gri,j}
{Nf is the number of Np-pixel frames (=$2^b$)}
fi,j = 0 for all i in [0, Nr], j in [0, Nc]
{One Granule counters per pixel}
k := 0
while k <= Nf-1 do {Current frame}
   for i in 0 to Nr-1 do {Current frame row}
```

```
      for j in 0 to Nc-1 do {Current frame
column}
         {granules counting}
         compute Gi,j(k+1)
         fi,j := fi,j + Gi,j(k+1)
         Gri,j:= fi,j/(k+1)
      end for
   end for
end while
```

D. *Memory requirements*

- Granules counters

$N_p = (N_r \text{ x } N_c)$ granules counters are needed to cope which every pixel activity monitoring. This involves a memory requirement of

$$G_{count} = N_p \times L \tag{10}$$

where L stands for the register-cell size. As the maximum quantity of instances (frames) is $Nf = 2^b$, b bits are required for L. Actually, b-1 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 set-up

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

$$H_{count} = 256 \text{ x } \log_2 N_p \tag{11}$$

- Zone-code memory flags

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

$$F_{mem} = R \text{ x } N_p \tag{12}$$

where R stands for the quantity of main intensity zones.

IV. Performances and time comparisons

High level software have been used to implement the fuzzy granular algorithm; a typical average time to process 128 instances for 6 THSP samples, using ***Matlab***® software in a personal desktop computer, is 0.01 sec., as reported in [5]. As an order of magnitude, one can assume that high level software's would process 100 instances of (512 x 512)-frames, in 500 to 600 sec. range time delays.

Table 1 shows the experimental results carried out on FPGA platform Xilinx® Virtex-6 (xc6vlx130t ff484 -3). The first column has image sizes. The minimum time 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. Considering a latency of 10 clock cycles for 8-bit precision activity indexes, it is clear that this implementation can be used for very challenging real-time applications.

	#inst	64	128
128x128	t_p (ns)	5.07	5.67
	#ff	331	331
	#LUT	386	396
	#RAMB	4+3	5+2
256x256	t_p (ns)	6.79	7.49
	#ff	319	339
	#LUT	405	416
	#RAMB	20+1	22+1
512x512	t_p (ns)	10.05	10.34
	#ff	329	349
	#LUT	412	426
	#RAMB	80+1	88+1

Table 1. FPGA Results

Note that the required memory for the biggest images consumes only 33% of available resources in the selected device, one of the smallest in the Virtex-6 family. Although area and time results enable real-time applications, significant progress is expected through optimization of algorithms and related implementations.

V. Conclusions

The main feature of the Fuzzy Granular algorithm is the smaller amount of image sequences required to process the THSP compared with other methodologies [5]. As a field of application, this fast embedded algorithm could be addressed to the analysis of drying time of specific film coatings in an automated manufacturing process.

Furthermore for other applications, where time could be a critical factor, real time processing could be a must. For this purpose, FPGA implementations provide low cost handy solutions whose performances could be a key feature. The concepts of this paper will be explored further according to some specific applications. Within precise sets of constraints, explicit values of hardware and time consumption will be readily defined.

VI. Acknowledgements

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

VII. References

[1] Amalvy, J., Lasquibar, C., Arizaga, R., Rabal, H., Trivi, M. (2001), Application of dynamic speckle interferometry to the drying of coatings, Prog. Org. Coat. 42, 89-99.

[2] Arizaga, R,. Cap N., Rabal, H. J., Trivi, M. (2002). Display of local activity using dynamical speckle patterns. Optical Engineering, 41, 287-294.

[3] Arizaga, R., Grumel E., Cap, N., Trivi, M., Amalvy, J., Yepes, B., Ricaurte, G. (2006), Following the drying of spray paints using space and time contrast of dynamic speckle, J. Coat. Technol. 3(4), 295-299.

[4] Faccia, P., Pardini, O., Amalvy, J., Cap N., Grumel, E., Arizaga, R. and Trivi M. (2008) Differentiation of the drying time of paints by dynamic speckle interferometry. Prog.Org.Coat. In Press doi:10.1016/j.porgcoat.2008.07.016

[5] Dai Pra, A., Passoni, L., Rabal, H. (2009), Evaluation of laser dynamic speckle signals applying granular computing, Signal Process. 89 (2009) 266–274.

[6] Dai Pra, A., Passoni, L., Rabal, H. (2009), Fuzzy Granular computing and Dynamic Speckle Interferometry for the identification of different thickness of wet coatings, Infocomp Journal of computer science, vol. 8-4, dec. 2009, pp.45-51.

[7] Brunel, L., Brun, A., Snabre, P. (2006) Microstructure movements study by dynamic speckle analysis. In Speckle'06: Speckles, from grains to flowers, edited by P.Slangen, C. Cerruti. Proc. of SPIE. 6341 doi 10-1117/12695493.

[8] Xilinx Inc., Virtex-6 user guide, http://www.xilinx.com, 2010.

[9] Murialdo Silvia E., Gonzalo H. Sendra, Lucía I. Passoni, Ricardo Arizaga, J. Froilán Gonzalez, Héctor Rabal and Marcelo Trivi, "Analysis of bacterial chemotactic response using dynamic laser speckle", J. Biomed. Opt. 14, 064015 (Nov 19, 2009); doi:10.1117/1.3262608

[10] Braga, R., Dal Fabbro, I., Borem, F., Rabelo, G., Arizaga, R., Rabal, H., and Trivi, M. Assessment of Seed Viability by Laser Speckle Techniques. Biosystems Engineering. v. 86(3), p. 287-294, 2003.

[11] Pajuelo, M., Baldwin, G., Rabal, H. Cap, N., Arizaga, R., Trivi, M. Biospeckle assessment of bruising in fruits, Optics and Laser in Engineering, v.40, p.13-24, 2003.

[12] Oulamara, A., Tribillon, G. , Duvernoy, J. Biological activity measurements on botanical specimen surfaces using a temporal de-correlation effect of laser speckle, Journal of Modern Optics. v. 36, p. 165-179, 1989.

[13] Rabal H. & Braga Jr. R. Dynamic Laser Speckle and Applications. CRC Press. 2008.

[14] Pedrycz, W. (2001) Granular computing: An emerging Paradigm, Physica-Verlag.

[15] Zadeh, L.A. (1970).Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic" Fuzzy sets and system, 90, 111-127

[16] Zadeh,L.A.(1965). Fuzzy Sets. Information and Control, 8, 338-353