Decision Tree Based FPGA-Architecture for Texture Sea State Classification

Santos López-Estrada, René Cumplido
Computer Science Department, National Institute for Astrophysics, Optics and Electronics
P.O.Box 51 and 216, Puebla, 72000, México
santosle@inaoep.mx, rcumplido@inaoep.mx.

Abstract

The target detection process in sea clutter background involves the use of different types of CFAR (Constant False Alarm Rate) algorithms. These algorithms and their parameters should be configured to obtain the maximum detection probability and minimum false alarm probability at the current sea state (Beaufort scale). This paper present an FPGA-architecture for automatic classification based on texture recognition of sea states. The sea state texture classification will allow select the appropriate CFAR algorithm and its parameters for the target detection process. The paper is centered in the hardware implementation for sea state texture classification, based on decision tree. The rules for decision tree are obtained from the analysis of the grey levels co-occurrence matrix features applied in an image of the sea state obtained in a radar scan. Results with simulated and real data are presented and discussed.

1. Introduction

The target detection process in background sea clutter can be automated by means of a detection processor known as constant false alarm rate processor (CFAR processor) [1]. This detection processor is based on statistical models with known parameters that are proportional to the expected magnitude power of radar echoes. The envelope of radar echoes are digitalized at different sample rates, each sample is named range cell. The detection processor calculates a threshold adaptively based on a local noise power of a group of reference cells surrounding a cell under test.

Detection algorithms have been developed with adaptive threshold in order to maintain a low level constant false alarm rate and high probability of detection. Such is the case of the classical Cell Averaging (CA) CFAR processor, which sets adaptively the threshold estimating the mean level in a window of M range cells. Several modifications have been proposed to improve their performance for different sea clutter conditions. Among the main variations of CA-CFAR algorithm are Greatest Of (GO-CFAR), Smallest Of (SO-CFAR), and Order Statistics (OS-CFAR) [1]. Each one of these CFAR variants solves the problems associated to sea clutter power transition only if the threshold calculation is obtained with a priori Knowledge of the current sea clutter type (Beaufort scale[1]). The main problem is to determine at which clutter type belong the new radar scan data, in order to associate a statistical model with CFAR algorithm to maintain a high performance in target detection process.

To solve this problem, a texture analysis method can be used to classify remote sensed images [2]. In our case, sea texture recognition to obtain a new classification of twelve sea state. The image texture is defined as a function of the spatial variations in grey values pixel intensities; we can extend this definition to radar signals as power level variations in return echoes. The spatial distribution provides a significant amount of information for the interpretation of the peak waves distribution in sea clutter, this information is interpreted as a sea texture [3]. The texture classification involves to decide which of the twelve texture categories to observed radar scan belongs to. For this classification a priori knowledge of the classes to be recognized is needed. At this point we have used the sea clutter model reported in [4], [5]. With this knowledge we carry out the texture analysis of each one model that describes the sea state levels. The grey level co-occurrence matrix (GLCM) and its features are obtained for twelve classes defined in the problem. As an off-line work we have obtained texture features from a set of radar images and we have used this information as a training set for different classifiers to obtain a classification based on decision tree. Weka (Waikato Environment for Knowledge Analysis) tool was used for data processing [6].

The main purpose for the FPGA-architecture is to calculate the GLCM and extract texture features to evaluate the decision tree and provide the classification for the sea state. The calculations of texture analysis were

[1] The Beaufort scale is a twelve sea state classification based on the wave height and wind speed

1-4244-0690-0/06/$20.00 ©2006 IEEE.

elaborated in fix point arithmetic to reduce calculation complexity and memory space.

The content of this paper is as follows: Section 2 describes the related work in texture analysis and architectures. Section 3 presents the methods for the sea state texture recognition. Section 4 provides a description of proposed FPGA-architecture. Section 5 presents some results obtained, which are discussed. Finally, some conclusions are presented in section 6.

2. Related Work

The pattern obtained by the reflectance of electromagnetic waves over the sea surface can be obtained by texture analysis. In these sense Ruiz *et al* [7] report the effectiveness of second order statistics by means of GLCM features, for applications where the space distribution of grey levels or power levels is important, such as in radar signals. Texture procedures require a great quantity of storage and a high computational cost when matrices are manipulated sequentially. To solve the problem many works have proposed FPGA-DSP architectures for calculation increase and take advantage of the parallelism in texture algorithms. *Ye et al* [8] they stand out those procedural textures can be effectively used to enhance the visual realism of 2D and 3D images. They proposed an FPGA-DSP architecture for analysis texture of images on the fly; with this method they reduce the storage. *Picó et al* [9] proposes a texture analysis co-processor. This work shows that second order statistics analysis does not necessarily imply a complex and expensive implementation in order to obtain good results in texture analysis. They propose hybrid FPGA-DSP architecture in floating point arithmetic to calculate texture features, but only in DSP module. In [10] Bariamis et al propose an FPGA-DSP architecture for Image texture extraction. They propose fix point arithmetic calculations in FPGA and floating point calculations in DSP.

Pezzuol et al in [11] propose a method to reduce and transform a decision tree into binary decision tree in order to facilitate FPGA synthesis.

3. Methods

In this section, the used data characteristics and methodology for texture classification are presented. Spatial data distribution and spatial dependence in radar scan images deserve special attention. These properties are essential for classification performance.

3.1 Data sets

The images used in this work have been generated with synthetic and real data. With simulated data have been generated sea clutter images for each sea state, each one with a clutter statistical model as a Weibull and K with different parameters. These data have been processed to present similar characteristics to real data. That is 8 bits resolution or 256 grey levels and mean equivalent to termic noise of 50mV. The size of all images is 512x512 pixels.

The real data have been generated with non-coherent X band radar with ten meters of range resolution. The returned echoes have been digitalized at 8 bits or 256 grey levels at 100MHz. The row data make a 4096 x 4096 matrix for each radar scan (radar image). An image of 512 x 512 is extracted in each radar scan. The real images for this work correspond only to sea state 1 to 4. The figure 1 shows textures of real images and texture details of the surface.

3.2 Co-occurrence features

For classification of sea texture we calculate 8 statistical features derived of grey levels co-occurrence matrix (GLCM), based on the features proposed by Haralik [12]. One of the defining qualities of texture is the spatial distribution of grey levels, in our domain, the power level of radar signal. The GLCM describes the spatial distribution of data in a window of radar scan.

The GLCM estimates image properties related to second-order statistics, considering the spatial relation between reference pixel and the neighboring pixel. This spatial relation can be represented by a displacement vector d = (dx,dy) [2]. The GLCM can represent displacements in 4 directions named 0^o, 45^o, 90^o and 135^o as shown in figure 2. For our domain, only 0^o and 90^o direction are analyzed. These directions are equivalents to range and azimuth in radar image. The vector distance used is dx, dy=1,2,3,4 due to radar resolution.

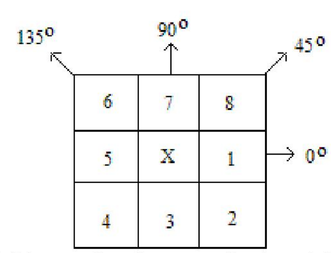

Fig. 2. Neighbors of reference pixel and displacement directions.

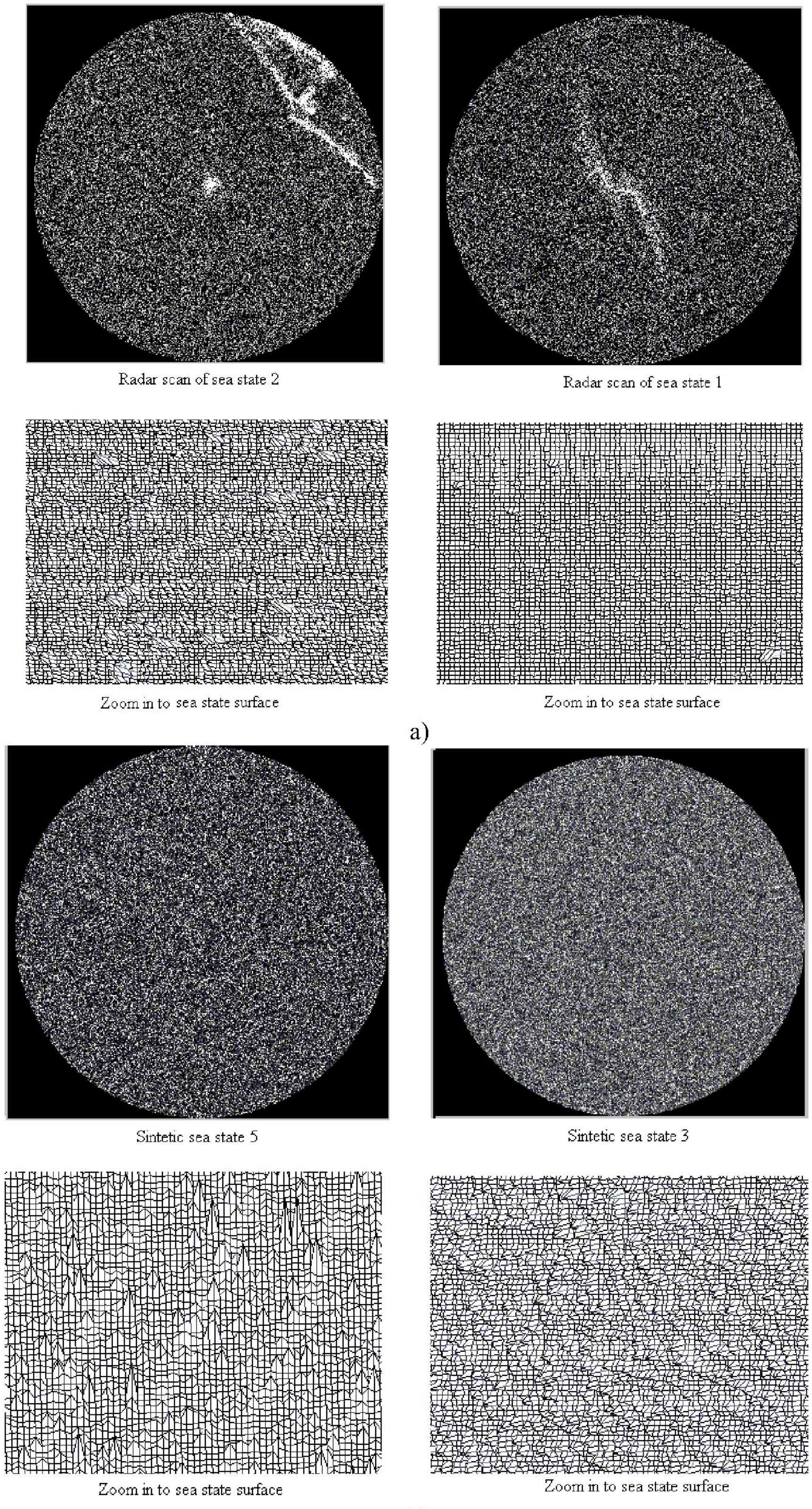

Fig. 1. a)Real, b) Synthetic radar scan and zoom-in to surface texture.

The GLCM is applied on N x N image, N=512, The image is defined as {I(x,y), 0≤x≤N-1, 0≤y≤N-1} with G=256 grey levels. The G x G GLCM denoted by $\mathbf{P_d}$ for a displacement vector **d** = (dx,dy) is defined as shown in the equation 1. The entry i, j of $\mathbf{P_d}$ is the number of occurrences of the pair of gray levels i and j which are a distance **d** apart.

$$P_d(i,j) = \{((r,s),(t,v)) : I(r,s) = i, I((t,v)) = j\} \quad (1)$$

Where:
(r,s), (t,v) ∈ N x N
(t,v)=(r+dx,s+dy)

The off-line work on the texture classification showed that the features that better classify the sea states are those that were implemented in the FPGA and they are described next

The *dissimilarity* is a measure of local variation of an image. It is defined as:

$$Dissimilarity = \sum_{i,j=0}^{n-1} P_{i,j} \, |i-j| \quad (2)$$

Two important features that provide good results in the analysis of grey levels [13] are the moments. Equations 3 and 4 represent the *angular second moment* (ASM) and *inverse difference moment* (MDI).

$$ASM = \sum_{i,j=0}^{n-1} P_{i,j}^{\;2} \quad (3)$$

$$MDI = \sum_{i,j=0}^{n-1} \frac{P_{i,j}}{1+(i-j)^2} \quad (4)$$

In all equations *Pij* represent the GLCM normalized, this is obtained dividing each entry Vij by sum of all entries Vij as show in equation 5.

$$P_{i,j} = \frac{V_{i,j}}{\sum_{i,j=0}^{n-1} V_{i,j}} \quad (5)$$

To carry out an implementation on the fix point arithmetic, we have transformed the equations 2, 3, 4 like it is shown next

$$Dissimilarity = \frac{1}{Vn} \sum_{i,j=0}^{n-1} V_{i,j} \, |i-j| \quad (6)$$

$$ASM = \frac{1}{Vn^2} \sum_{i,j=0}^{n-1} V_{i,j}^{\;2} \quad (7)$$

$$MDI = \frac{1}{Vn} \sum_{i,j=0}^{n-1} \frac{V_{i,j}}{1+(i-j)^2} \quad (8)$$

Where:
1/Vn, $1/Vn^2$ represent integer numbers for normalization.
$1/1+(i-j)^2$ is pre-calculated and stored in a LUT

The equations 5. 6, 7, 8 are used for obtain the GLCM and texture features with integer values. With this features the next step is evaluate the decision tree obtained in the off-line work. We have obtained decision tree for simulated and real data. For real data the percent of classification obtained with training data is 90.47% using MDI and dissimilarity features and for test data is 90%, both using the classification algorithm C4.5[6]. Figure 3 show the confusion matrix, it can be observed that wrong classified sea states has been classified as sea state with one level of difference. Figure 4 show the decision tree for real data, similar trees has been obtained for synthetic data.

a)

Correctly Classified Instances	19	90.4761 %
Incorrectly Classified Instances	2	9.5238 %
Kappa statistic	0.8828	

```
a b c d   <-- classified as
9 1 0 0 | a = ST1
0 4 1 0 | b = ST2
0 0 3 0 | c = ST3
0 0 0 3 | d = ST4
```

b)

Correctly Classified Instances	9	90 %
Incorrectly Classified Instances	1	10 %
Kappa statistic	0.8611	

```
a b c d   <-- classified as
4 0 0 0 | a = ST1
0 1 1 0 | b = ST2
0 0 2 0 | c = ST3
0 0 0 2 | d = ST4
```

Fig. 3. Confusion matrix a) training set b) test set.

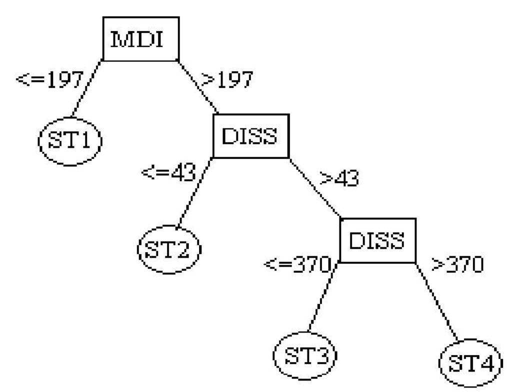

Fig. 4. Decision tree for real data.

4. Architecture description

The proposed architecture consists of two stages: texture analysis and texture classification. In the first stage the radar image is analyzed, the GLCM with integer values is obtained. The features textures are extracted of this matrix, this information is processed for a second stage, which is a decision tree for the texture classification. Figure 5 show a block diagram of the proposed architecture.

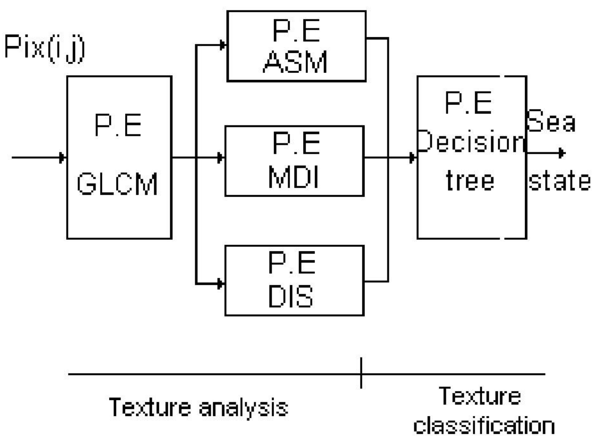

Fig. 5. Proposed architecture.

4.1 Texture analysis

The first stage of the proposed architecture named texture analysis is composed by 4 processor elements (PE) named PEGLCM, PEASM, PEMDI, PEDIS. The first PE calculate the GLCM with integer values, this process is carried out by a set of accuulator that counts the grey levels in a neighborhood. Figure 6 show a block diagram for this PE. The input for this PE is a pixels of the radar image, the value of the pixel(i,j) is the row index while pixel (i,j+d) is the column index for the selected accumulator to increase in 1. At the final process the accumulators contain the GLCM values

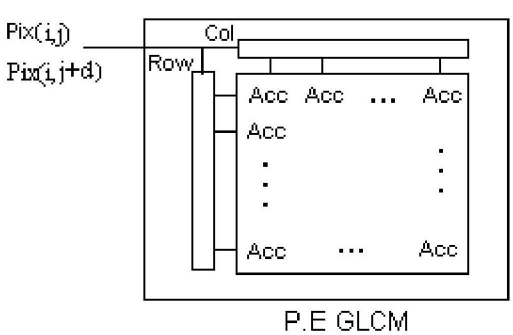

Fig. 6. Processor element for the GLCM

The angular second moment feature texture is obtained processing the GLCM data with equation 7. Figure 7 show the block diagram for this processor element. The input for this PE is Vij that represent the element i,j of the GLCM and Vn^2 that represent the normalization value.

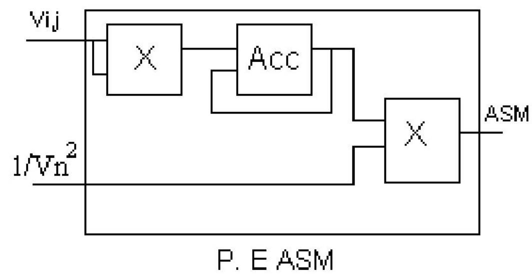

Fig. 7. Processor element for ASM

Figure 8 show the processor element for the inverse difference moment, this feature is obtained processing the GLCM data with equation 8. The input data is Vij and 1/Vn discussed in the previous section. The elements i, j select the appropriate value in the LUT for the expression $1/1+(i-j)^2$.

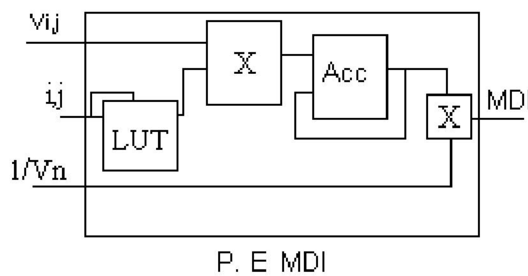

Fig. 8. Processor element for MDI

Finally the dissimilarity is calculated with equation 6 and a block diagram is showed in figure 9 where shows the data input for obtain dissimilarity.

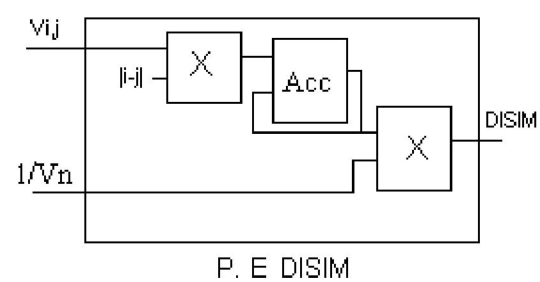

Fig. 9. Processor element for dissimilarity

The latency in this stage depends only of the size image to process, since the PEASM, PEMDI and PEDISIM can process data until the PEGLCM finalize the construction of the GLCM. For this experiment we have used radar images of 128x 128 pixels and 256x256 pixels with 8 bits or 256 grey levels.

4.2 Texture classification

The second stage of the proposed architecture evaluates a decision tree obtained in off-line work described in section 3. This stage consists of one PE whose block diagram is showed in figure 10. The implementation of decision tree is a module that change constantly and depend of the process of classification, therefore this module should be reconfigurable. For this experiment only have been implemented two decision tree, for simulated and real data. Figure 10 show the implementation of decision tree showed in figure 4

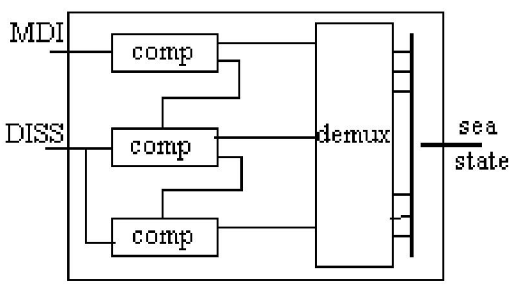

Fig. 10. Processor element for texture classification

5. Results and discussion

The architecture was implemented in a FPGA of family Virtex2 xc2v250-6-fg256 in VHDL language. Some statistical data of the implementation are sowed next.

The Design Statistics

# IOs	: 132
# Registers	: 5
# Adders/Subtractors	: 3
# Multipliers	: 7
# BELS	: 235
# FlipFlops/Latches	: 116
# Clock Buffers	: 1
# IO Buffers	: 131
# MULTs	: 9

Selected Device : 2v250fg256-6

Number of Slices:	81	out of	1536	5%
Number of Slice Flip Flops:	116	out of	3072	3%
Number of 4 input LUTs:	80	out of	3072	2%
Number of bonded IOBs:	132	out of	172	76%
Number of MULT18X18s:	9	out of	24	37%
Number of GCLKs:	1	out of	16	6%

Timing Summary:

Minimum period: 6.975ns (Maximum Frequency: 143.369MHz)
Minimum input arrival time before clock: 7.247ns
Maximum output required time after clock: 6.780ns

The maximum frequency is 143 MHz but at the moment the maximum frequency for the acquisition data is 100MHz. The image radar scan must be acquired and proceced in 2.5 seconds therefore the implementation can be process data in real time.

The extraction of GLCM requires scan all image therefore the maximum latency depend only of the image size. In the current work the image size tested was 16x16, 128x128 and 256x256 pixels with 16, 128 and 256 grey levels respectively

The results obtained were validated with an application developed in matlab working with floating point arithmetic. The comparison with results obtained with proposed architecture shows that the maximum error obtained is 0.46% in the worst case. This error is compatible with the error in confusion matrix, generating a bad classification only in one level of the sea state.

6. Conclusions and future work

We have proposed an architecture to carry out two fundamental tasks. Fisrt, a textural analysis based on the co-ocurrences matrix. This calculations in this architecture are in fixed point arithmetic reducing the computational complexity and the problem of storage, processing the images on the fly, improving the error reported by Barianis in [10]. Second, the classification of sea texture based on decision tree. This module must be modified according to decision tree obtained in an off-line work, therefore is a good module for reconfiguration. The results obtained sows that the architecture is a good classifier for sea states. As future work we are developing a modulate to calculate the energy analysis using the energy features of Lews to increase the 90% of the classification showed in the confusion matrix of the figure 3.

References

[1]. Skolnik M. “Introduction to radar system”. Mc Graw Hill. New York. 2001.

[2]. Chen C. H, Pau L.F. :”The Handbook of Pattern Recognition and Computer Vision”. 2nd edn. World Scientific Publishing Co. 1998 207-248.

[3]. Centeno J., Weindorf M.:”Spatial Information for Image Segmentation”. CIPA International Symposium. 1999

[4]. Farina, Gini, Greco. "High Resolution Sea Clutter Data: A Statistical Analysis of Recorded Live Data" IEE Proceedings Radar, Sonar & Navigation. Vol. 144 No. 3. 1997. pp 121-130

[5]. S. Lopez, R. Cumplido “A Hybrid Approach for Target Detection Using CFAR Algorithm and Image Processing” Fifth Mexican International Conference on Computer Science. 2004

[6]. Witten I. H., Eibe F.: “Data Mining, Practical Machine Learning Tools and Techniques”. 2nd edn. Morgan Kaufman Publishers.2005

[7]. Ruiz L., Sarria A., Recio J.: “Evaluation of Texture Analysis Techniques to Characterize Vegetation”. Advances in Qantitative Remote Sensing. 2002. 514-521.

[8]. Andy G. Ye, David M. Lewis. “Procedural Texture Mapping on FPGAs” International Symposium on Field Programmable Gate Arrays. 1999 Pp 112-120

[9]. Ibarra Picó, Cuenca Asensi, Córcoles V.: “Accelerating Statistical Texture Analysis with an FPGA-DSP Hybrid Architecture”. Proceedings IEEE Symposium on Field-Programmable Custom Computing Machines. 2001

[10]. Barianis D. G., Iakovidis D.” An FPGA-based Architecture for Real Time Image Feature Extraction” International Conference on Pattern Recognition 2004 pp. 801-804

[11]. Pezzuol R. “Multiplexor-Based FPGA Synthesis”

[12]. Haralick R. M., Shammugam K.: Texture Features for image classification. IEEE transaction on systems, Man and Cybernetics. 1973. 610-622.

[13]. Tuceryan M.:Moment Based Texture Segmentation. Pattern Recognition Letters vol. 15 1994. 659-668.