Gray Level Co-Occurrence Matrix Computation Based On Haar Wavelet

M. M. Mokji[1], S.A.R. Abu Bakar[2]
Faculty of Electrical Engineering
University of Technology Malaysia, Malaysia
[1]musa@fke.utm.my, [2]syed@fke.utm.my

Abstract

In this paper, a new computation for gray level co-occurrence matrix (GLCM) is proposed. The aim is to reduce the computation burden of the original GLCM computation. The proposed computation will be based on Haar wavelet transform. Haar wavelet transform is chosen because the resulting wavelet bands are strongly correlated with the orientation elements in the GLCM computation. The second reason is because the total pixel entries for Haar wavelet transform is always minimum. Thus, the GLCM computation burden can be reduced. The proposed computation is tested with the classification performance of the Brodatz texture images. Although the aim is to achieve at least similar performance with the original GLCM computation, the proposed computation gives a slightly better performance compare to the original GLCM computation.

1. Introduction

Gray level co-occurrence matrix (GLCM) has been proven to be a very powerful tool for texture image segmentation [1, 2]. The only shortcoming of the GLCM is its computational cost. Such restriction causes impractical implementation for pixel-by-pixel image processing. In the previous works, GLCM computational burden was reduced by two methods, at the computation architecture level and hardware level. D. A. Clausi *et. al.* restructures the GLCM by introducing a GLCLL (gray level co-occurrence linked list), which discard the zero value in the GLCM [3]. This technique gives a good improvement because most GLCM is a sparse matrix where most of its values are equals to zero. Thus the size of GLCLL is significantly smaller than GLCM. Then the structure of the GLCLL was improved in [4, 5]. Another work is presented in [6] where fast calculation of GLCM texture features relative to a window spanning an image in a raster manner was introduced. This technique was based on the fact that windows relative to adjacent pixels are mostly overlapping, thus the features related to the pixels inside the overlapping windows can be obtained by updating the early-calculated values. In January 2007, S. Kiranyaz and M. Gabbouj proposed a novel indexing technique called Hierarchical Cellular Tree (HCT) to handle large data [7]. In his work, it was proved that the proposed technique is able to reduce the GLCM texture features computation burden.

At the hardware level, paper presented in [8] implemented the GLCM texture features computation using FPGA. In this work, the computation speed using the FPGA produced a better result when compared with the computation on a general-purpose processor (Pentium 4 with 2400MHz clock speed). The architecture of the FPGA is then improved in [9] for more efficient computation.

In this work, a new GLCM computation is proposed in order to reduce its computational burden. The GLCM computation will be based on Haar wavelet transform. The subsequent topics will be divided into 4 sections. The first and second sections will discuss briefly on the introduction of GLCM and Haar wavelet transform respectively. The third section will discuss the proposed GLCM computation, which is based on Haar wavelet transform. In the last section, the performance of the proposed technique is measured and compared with the original computation.

2. Gray Level Co-Occurrence Matrix

GLCM is a matrix that describes the frequency of one gray level appearing in a specified spatial linear relationship with another gray level within the area of investigation [10]. Here, the co-occurrence matrix is computed based on two parameters, which are the relative distance between the pixel pair d measured in pixel number and their relative orientation ϕ.

Computer Graphics, Imaging and Visualisation (CGIV 2007)
0-7695-2928-3/07 $25.00 © 2007 IEEE

Normally, ϕ is quantized in four directions (*0^0, 45^0, 90^0 and 135^0*) [10]. In practice, for each d, the resulting values for the four directions are averaged out. To show how the computation is done, for image I, let m represent the gray level of pixels (x, y) and n represent the gray level of pixels $(x \pm d\phi_0, y \pm d\phi_1)$ with L level of gray tones where $0 \le x \le M-1$, $0 \le y \le N-1$ and $0 \le m,n \le L-1$. From these representations, the gray level co-occurrence matrix $C_{m,n}$ for distance d and direction ϕ can be defined as

$$C_{m,n,\phi} = \sum_x \sum_y P\{I(x,y) = m \;\&\; I(x \pm d\phi_0, y \mp d\phi_1) = n\} \tag{1}$$

where $P\{.\} = 1$ if the argument is true and otherwise, $P\{.\} = 0$. For each ϕ value, its ϕ_0 and ϕ_1 values are referred as in the Table 1. One of the characteristic of the GLCM is, it is diagonally symmetry where $C_{m,n} = C_{n,m}$. Thus, the computation of the GLCM can be simplified as in Equation 2. Now, the $\pm$ and $\mp$ signs are replaced with single operation of $+$ and – accordingly. As compensation, the resulting $C_{m,n}$ is added with $C_{n,m}$ to obtain a complete GLCM. For the rest of this paper, GLCM computation will be referred to the method as in Equation 2.

$$C_{m,n,\phi} = \sum_x \sum_y P\{I(x,y) = m \;\&\; I(x + d\phi_0, y - d\phi_1) = n\} \tag{2}$$

Table 1: Orientation constant

ϕ	ϕ_0	ϕ_1
0^0	0	1
45^0	-1	-1
90^0	1	0
135^0	1	-1

In the classical paper [11], Haralick *et. al* introduced fourteen textural features from the GLCM and then in [12] stated that only six of the textural features are considered to be the most relevant. Those textural features are *Energy, Entropy, Contrast, Variance, Correlation and Inverse Difference Moment*. *Energy* is also called *Angular Second Moment* (ASM) where it measures textural uniformity [10]. If an image is completely homogeneous, its *energy* will be maximum. *Entropy* is a measure, which is inversely correlated to *energy*. It measures the disorder or randomness of an image [10]. Next, *contrast* is a measure of local gray level variation of an image. This parameter takes low value for a smooth image and high value for a coarse image. On the other hand, *inverse difference moment* is a measure that takes a high value for a low contrast image. Thus, the parameter is more sensitive to the presence of the GLCM elements, which are nearer to the symmetry line $C(m,m)$ [10]. Variance as the fifth parameter is a measure that is similar to the first order statistical variables called standard deviation [13]. The last parameter, *correlation*, measures the linear dependency among neighboring pixels. It gives a measure of abrupt pixel transitions in the image [14].

3. Haar Wavelet Transform

An image that undergoes Haar wavelet transform will be divided into four bands at each of the transform level [15]. The first band represents the input image filtered with a low pass filter and compressed to half. This band is also called 'approximation'. The other three bands are called 'details' where high pass filter is applied. These bands contain directional characteristics. The size of each of the bands is also compressed to half. Specifically, the second band contains vertical characteristics, the third band shows characteristics in the horizontal direction and the last band represents diagonal characteristics of the input image.

Conceptually, Haar wavelet is very simple because it is constructed from a square wave, which is represented by Equation 3 and 4 [16]. Moreover, Haar wavelet computation is fast since it only contains two coefficients and it does not need a temporary array for multi-level transformation [17]. Thus, each pixel in an image that will go through the wavelet transform computation will be used only once and no pixel overlapping during the computation. Theoretically, this characteristic can be used to reduce the GLCM computation, which will be discussed in the subsequent topic.

$$\varphi_L(n) = \begin{cases} \frac{1}{\sqrt{2}} & n = 2k, 2k+1 \\ 0 & elsewhere \end{cases} \tag{3}$$

$$\varphi_H(n) = \begin{cases} \frac{1}{\sqrt{2}} & n = 2k \\ -\frac{1}{\sqrt{2}} & n = 2k+1 \\ 0 & elsewhere \end{cases} \qquad (4)$$

According to Equation 3 and 4, the Haar basis only takes even-indexed pixel coordinates. The basis is also scaled by the factor of $\sqrt{2}$ so that it becomes orthonormal for a two points length basis. If the length of the basis is changed, the factorizing value will be different [16]. This subject matter will be discussed again later and for now, the computation will be focused on a two points Haar basis. From the Haar basis, wavelet transform of an image can be written in four different linear equations representing the four bands. Because image is a two-dimensional data, performing wavelet transform is done twice in each of its level. First, it is done at row wise and then at column wise. As Haar wavelet transform computes only on even-indexed coordinates and it has only two coefficients, this is why the transformation computation does not overlap on any image pixels.

4. Haar Wavelet Based GLCM Computation

4.1. Averaging and Differencing Formulation

Besides the typical way of writing the GLCM equation based on each image pixel and its desired neighboring pixels as in Equation 2, another interpretation based on averaging and differencing filter is also possible. These filters come into the picture because they are the same filter types that construct the Haar basis [17]. Thus, formulating the GLCM computation using the averaging and differencing filters will be the turning point of formulating the GLCM computation based on the Haar wavelet transform.

First, let $I(x,y) = \alpha$ and $I(x + d\phi_0, y - d\phi_1) = \beta$. As Haar basis is represented by Equation 3 and 4, then a simple two point length averaging and differencing basis can be written as

$$ave\,(n) = \begin{cases} \frac{1}{2} & n = k,\, k+1 \\ 0 & elsewhere \end{cases} \qquad (5)$$

$$dif\,(n) = \begin{cases} \frac{1}{2} & n = k \\ -\frac{1}{2} & n = k+1 \\ 0 & elsewhere \end{cases} \qquad (6)$$

Manipulating these averaging and differencing bases for the GLCM computation leads to four formulations for each of the averaging and differencing filters. This is because the GLCM computation is engaged with four orientation of ϕ. These computations are simplified by Equation 7 and 8 below.

$$L_\phi(x,y) = \frac{1}{2}I(x,y) \quad + \quad \frac{1}{2}I(x + d\phi_0, y - d\phi_1) \qquad (7)$$

$$H_\phi(x,y) = \frac{1}{2}I(x,y) \quad - \quad \frac{1}{2}I(x + d\phi_0, y - d\phi_1) \qquad (8)$$

Here, averaging and differencing operations are represented by $L(x,y)$ and $H(x,y)$ respectively instead of $ave(x,y)$ and $dif(x,y)$ to maintain the symbol consistency for the low pass filter and high pass filter. To write the GLCM formulation based on the averaging and differencing filters, the inverse of Equation 7 and 8 are computed to get the α and β. Equation 9 and 10 show the inverse operation while Equation 11 shows the GLCM computation based on the averaging and differencing filters.

$$I(x,y) = \alpha = L_\phi(x,y) + H_\phi(x,y) \qquad (9)$$

$$I(x + d\phi_0, y - d\phi_1) = \beta = L_\phi(x,y) - H_\phi(x,y) \qquad (10)$$

$$\begin{aligned} C_{m,n,\phi} = \sum_x \sum_y P\{ & L_\phi(x,y) + H_\phi(x,y) = m \\ & \& \;\; L_\phi(x,y) - H_\phi(x,y) = n\} \end{aligned} \qquad (11)$$

4.2. Haar Wavelet Based GLCM

Now, as GLCM computation has been written based on low pass (averaging) filter (Equation 7) and high pass (differencing) filter (Equation 8), it is easier to formulate it using the Haar basis. This is done by replacing the low pass filter and the high pass filter with the four bands of the Haar wavelet transform. Table 2 lists the filter replacements.

Table 2: Filter replacements

Filter (original GLCM)	Replacement (wavelet bands)
$L_{\phi}(x,y)$	$LL(x',y')$
$H_{0^\circ}(x,y)$	$LH(x',y')$
$H_{90^\circ}(x,y)$	$HL(x',y')$
$H_{45^\circ}(x,y)$ and $H_{135^\circ}(x,y)$	$HH(x',y')$

Referring to Table 2, low pass filter for all orientation of the ϕ are replaced with single approximation band. This is because the approximation band is the only band that performs pure low pass filtering process. Then, $H_{0^\circ}(x,y)$ is replaced with $LH(x',y')$ as this band contains horizontal characteristics and $HL(x',y')$ is replaces $H_{90^\circ}(x,y)$ because it contains the vertical characteristics. For $H_{45^\circ}(x,y)$ and $H_{135^\circ}(x,y)$, the filters are replaced with single $HH(x',y')$ band. This is because Haar wavelet transform combines both diagonal orientations (45° and 135°) computation in a single band, which is the $HH(x',y')$. Hence, instead of four orientations computation in the original GLCM, Haar wavelet based computation of the GLCM will only compute in three orientations.

Computing the GLCM based on the Haar wavelet bands is inefficient, similarly when computing the original GLCM based on the averaging and differencing filters. This is because a temporary array is needed to store the results of the bands. Fortunately, as mention in the previous topic, Haar wavelet transform does not need the temporary array for its computation. Thus, instead of computing the GLCM based on the wavelet bands, the computation of the GLCM based on the Haar wavelet can be done directly from the input image. To show how it is done, lets rearrange the formulation for the horizontal orientation.

First, referring to Equation 9 and 10 and Table 2, α and β for the horizontal orientation can be written as

$$\alpha = LL(x',y') + LH(x',y') \tag{12}$$

$$\beta = LL(x',y') - LH(x',y') \tag{13}$$

Then, referring to Equation 7 and 8, α and β can be written directly from the input image as below

$$\alpha = \tfrac{1}{2}I(x,y) + \tfrac{1}{2}I(x,y+1) \tag{14}$$

$$\beta = \tfrac{1}{2}I(x+1,y) + \tfrac{1}{2}I(x+1,y+1) \tag{15}$$

To generalize the formulation, variable d (distance) is entered. This is shown in Table 3 where α and β are now indexed with ϕ to differentiate their orientation. The table also shows α and β for vertical and diagonal orientation. In Equation 14 and 15, d is equal to 1.

Table 3: α and β values based on the wavelet orientation

Wavelet band (ϕ)	α_{ϕ} and β_{ϕ}
Horizontal (hor)	$\alpha_{hor} = \tfrac{1}{2}I(x,y) + \tfrac{1}{2}I(x,y+1)$ $\beta_{hor} = \tfrac{1}{2}I(x+d,y) + \tfrac{1}{2}I(x+d,y+1)$
Vertical (ver)	$\alpha_{ver} = \tfrac{1}{2}I(x,y) + \tfrac{1}{2}I(x+1,y)$ $\beta_{ver} = \tfrac{1}{2}I(x,y+d) + \tfrac{1}{2}I(x+1,y+d)$
Diagonal (dia)	$\alpha_{dia} = \tfrac{1}{2}I(x,y) + \tfrac{1}{2}I(x+d,y+d)$ $\beta_{dia} = \tfrac{1}{2}I(x+d,y) + \tfrac{1}{2}I(x,y+d)$

From the definition of GLCM, d can take any value from 1 to M-1 or N-1, whichever is lower. In the Haar basis, as mentioned in the previous topic, changing d is equivalent by the means of changing the length of the Haar basis. Thus, scaling α_{ϕ} and β_{ϕ} by the factor of $\sqrt{2}$ or 2 for an image with $d > 1$ will not conserve the orthonomality of the Haar basis. However, orthonomality is not an issue for the GLCM computation because the inverse computation of the GLCM is not necessary. Although the orthonomality is not conserved, the Haar basis is still orthogonal, which keeps the Haar basis uncorrelated. Based on these arguments, the scaling factor for α_{ϕ} and β_{ϕ} is retained by the value of 2 for any possible value of d. Furthermore, this scaling value will actually average out the value of the pixels involved in each of the α_{ϕ}

and β_{ϕ} computation. Certainly, averaging process will preserve the gray level range of the image. Then, based on the α_{ϕ} and β_{ϕ}, GLCM formulation can be rewritten as Equation 10.

$$C'_{m,n,\phi} = \sum_{x'}\sum_{y'} P\{\alpha_{\phi} = m \ \ \& \ \ \beta_{\phi} = n\} \qquad (10)$$

4.3. Pixel Entries in GLCM Computation

One way of determining the computational speed of the GLCM is by calculating its total pixel entries. This computation counts the total of how frequent each pixel in an image is used to compute the GLCM. As total pixel entries is about counting the pixel involvement, its computation has a linear relationship with the image size, which can be computed according to Equation 16 to Equation 18 below [18].

$$\eta_0 = M \times (N-1) \times 2 \qquad (16)$$

$$\eta_{90} = (M-1) \times N \times 2 \qquad (17)$$

$$\eta_{45} = \eta_{135} = (M-1) \times (N-1) \times 2 \qquad (18)$$

M is the row length and N is the column length. Thus, as the size of the image is increased, the total pixel entries will also increase. For the GLCM computation based on the Haar wavelet, its total pixel entries computation is different. This is because the computation is applied only on the even-indexed coordinates (x', y') of the input image. Hence, the total pixel entries is represented by Equation 19. This equation is true for all of the three orientations in the Haar wavelet based computation.

$$\eta_w = M \times N \qquad (19)$$

From Equation 16 to 19, it is obvious that the total pixel entries for Haar wavelet based computation is less than the original computation of the GLCM. As an example, if the input image size is 10×10, the total pixel entries for the original GLCM computation will be 180+180+162+162=684 while the total pixel entries for Haar wavelet based computation is 100+100+100=300. In this example, the Haar wavelet based computation only consumes 43.9% pixel entries of the original GLCM computation, which is less than half. At minimum reduction, the proposed computation consumes 75% pixel entries of the original computation. For maximum reduction, it can reach until 37.5% where the image size is very large. This shows that larger image size will result in better reduction of the pixel entries.

5. Experimental Results

It has been proven that GLCM computation based on Haar wavelet reduces the computational cost in terms of the pixel entries. Now, the performance of the new formulation compared to the original computation will be investigated. To measure the performance, classification accuracy of 25 Brodatz texture images selected randomly from the Brodatz album [19] are computed. In the experiment, there are three sample sets with different image size. In the first sample set, each Brodatz texture image is divided into 4 similar size images of 320x320 (original image size is 640x640). Thus, the first sample set will have 100 images. Then, the second sample set will have 400 images where each Brodatz texture image is divided into 16 images with the size of 160x160. For the third sample set, each Brodatz texture image is divided into smaller images with the size of 128x128. Thus, total images for the third set is 625 images. Larger images have more texture attributes compared to the smaller images. Therefore, the three sample sets of different image sizes will test both the original and the proposed computation with different image conditions. Figure 1 shows the example of an image, which is divided accordingly to the three sample sets.

In the classification process, the average value for the six textural features of the GLCM are computed for each of the Brodatz texture images that have been divided accordingly to the three sample sets. The purpose of this computation is to train the samples. Thus, for each sample set, there are 25 sets of trained data (value of the six textural features) as there are 25 original Brodatz texture images. Then, for each sample set, each sample image is classified to the nearest trained data based on Euclidean distance. Basically, the GLCM is computed based on two parameters, which are d and ϕ. In this work, the experiment is repeated for three value of d from 1 to 3. Then for each value of d, the resulting matrixes for the four orientations are averaged out. The classification accuracy of the Haar wavelet based GLCM computation and the original GLCM computation are shown in Table 4.

The aim for the proposed GLCM computation is to achieve at least similar classification accuracy compared to the original GLCM computation. Fortunately, referring to Table 4, most of the results show a slight improvement in the proposed GLCM computation except for the image size of 128x128 with

d equals to 1. The results for the proposed GLCM computation are also very consistent through out all of the experiment conditions where they follow the results pattern of the original GLCM computation.

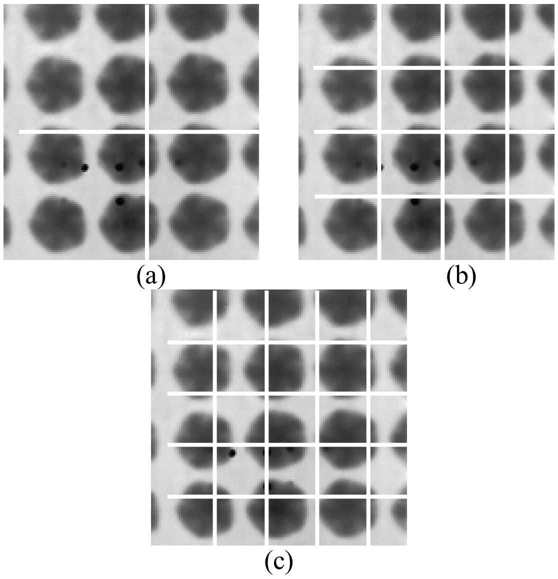
(a) (b) (c)

Figure 1: Original image division. (a) Sample set 1, (b) Sample set 2, (c) Sample set 3

Table 4: Classification results

Image size (MxN)	d	Classification Accuracy (%)	
		Original GLCM	Haar wavelet based GLCM
320x320	1	93	93
	2	88	91
	3	90	92
160x160	1	80.25	81.5
	2	75.25	80.25
	3	78.75	81.5
128x128	1	78.24	77.76
	2	74.72	79.36
	3	77.6	80.48

Classification accuracies for the Haar wavelet based GLCM are better compared to its original GLCM computation due to the smoothing process, which is embedded in the Haar wavelet computation. This can be shown in Equation 14 and 15 where α and β are the average (smoothing) of two pixel values. Thus, the image is enhanced when applied with smoothing. Smoothing is proven to be the factor that contributes to the increased classification accuracy as shown in Table 5. Results from Table 5 are obtained by repeating the earlier experiment using input images that have been filtered with an averaging filter. Table 5 shows that filtered input images give better classification accuracy compared to non-filtered input images. Therefore, with smoothing process embedded in the Haar wavelet based GLCM computation, it gives better classification accuracy compared to the original GLCM computation. Table 5 also shows the repeated experiment results for Haar wavelet based GLCM. In this case, the smoothing process is done twice, first applied at the input image and then in the GLCM computation. The results pattern is also similar with the earlier experiment as shown in Table 4, which means that the consistency of the results is conserved.

Table 5: Classification accuracy for filtered input image

Image size (MxN)	d	Classification Accuracy (%)		
		Original GLCM Without Smoothing	Original GLCM With Smoothing	Wavelet Based GLCM With Smoothing
320x320	1	93	94	94
	2	88	88	94
	3	90	87	91
160x160	1	80.25	83.75	84.75
	2	75.25	79	82.25
	3	78.75	79.25	79.5
128x128	1	78.24	80.16	79.72
	2	74.72	77.12	80
	3	77.6	75.52	79.2

6. Conclusion

This paper has presented a new technique for GLCM computation based on Haar wavelet transform. Computing the GLCM based on Haar wavelet transform has the ability to reduced the computational burden in terms of pixel entries up to 62.5% reduction. In terms of performance measurement, Haar wavelet transform does not only reduce the computational burden but also increase the classification accuracy of Brodatz texture images when compared to the original computation.

References

[1] J. Weszka, C. Dyer, A. Rosenfeld, “A Comparative Study Of Texture Measures For Terrain Classification” *IEEE Trans. SMC-6 (4)*, pp. 269-285, April 1976.

[2] R.W. Conners, C.A. Harlow, “A Theoretical Comparison Of Texture Algorithms”, *IEEE Trans. PAMI-2*, pp. 205-222, 1980.

[3] D.A. Clausi, M.E. Jernigan, “A fast method to determine co-occurrence texture features”, *IEEE Trans on Geoscience & Rem. Sens*., vol. 36(1), pp. 298-300, 1998.

[4] D.A. Clausi, Yongping Zhao, “An advanced computational method to determine co-occurrence probability texture features”, *IEEE Int. Geoscience and Rem. Sens. Sym,* vol. 4, pp. 2453-2455 2002.

[5] A.E. Svolos, A. Todd-Pokropek, “Time and space results of dynamic texture feature extraction in MR and CT image analysis”, *IEEE Trans. on Information Tech. in Biomedicine*, vol. 2(2), pp. 48-54, 1998.

[6] F. Argenti, L. Alparone, G. Benelli, “Fast algorithms for texture analysis using co-occurrence matrices”, *IEE Proc on Radar and Signal Processing*, vol. 137(6), pp. 443-448, 1990.

[7] S. Kiranyaz, M. Gabbouj, “Hierarchical Cellular Tree: An Efficient Indexing Scheme for Content-Based Retrieval on Multimedia Databases”, *IEEE Transactions on Multimedia*, vol. 9(1), pp. 102-119, 2007.

[8] M.A. Tahir, A. Bouridane, F. Kurugollu, A. Amira, “An FPGA based coprocessor for calculating Grey level co-occurrence matrix”, *IEEE Int. Midwest Symp. on Circuits and Systems*. vol. 2, pp. 868-871, 2003.

[9] M.A. Tahir, A. Bouridane, F. Kurugollu, A. Amira, “Accelerating the computation of GLCM and Haralick texture features on reconfigurable hardware”, *Int. Conf. on Image Processing*, vol. 5, pp 2857-2860, 2004.

[10] A. Baraldi, F. Parmiggiani, “An Investigation Of The Textural Characteristics Associated With GLCM Matrix Statistical Parameters”, *IEEE Trans. on Geos. and Rem. Sens*., vol. 33(2), pp. 293-304, 1995.

[11] R. Haralick, K. Shanmugam, I. Dinstein, “Texture Features For Image Classification”, *IEEE Transaction*, SMC-3(6). Pp. 610-621, 1973.

[12] N. Otsu, “A Threshold Selection Method from Gray-Level Histogram”, *IEEE Trans. on System Man Cybernetics,* vol. 9(I), pp. 62-66, 1979.

[13] P. Gong, J. D. Marceau, and P. J. Howarth, “A comparison of spatial feature extraction algorithms for land-use classification with SPOT HRV data”, *Proc. Remote Sensing Environ*, vol. 40, pp. 137-151, 1992.

[14] A. Ukovich, G. Impoco, G. Ramponi, “A tool based on the GLCM to measure the performance of dynamic range reduction algorithms”, *IEEE Int. Workshop on Imaging Sys. & Techniques*, pp. 36-41, 2005.

[15] C. Schremmer, “Decomposition strategies for wavelet-based image coding”, *Int. Symp. on Signal Processing and its Applications*, vol. 2, pp. 529-532, 2001.

[16] M. Vetterli, “Wavelets and Subband Coding”, *Prentice Hall PTR*, 1995.

[17] F.K.-P. Chan, A.W.-C. Fu, C. Yu, “Haar wavelets for efficient similarity search of time-series: with and without time warping”, *IEEE Trans. on Knowledge and Data Engineering*. vol. 15(3). Pp. 686-705, 2003.

[18] S. Theodoridis, K. Koutroumbas, “Pattern Recognition Second Edition”, *Elsevier, USA*, 2003.

[19] P. Brodatz, “Textures: a Photographic Album for Artists and Designers”, *Dover, New York*, 1965.