Local Similarities for Boosting the Performance of Local Binary Patterns Technique

Abdelhamid Abdesselam
Dept. of Computer Science, Sultan Qaboos University, Oman
Email: ahamid@squ.edu.om

***Abstract*—Texture analysis is a crucial step in various computer vision and pattern recognition applications. A large number of techniques for describing, and retrieving texture images has been proposed during the last few decades. The conventional Local Binary Patterns (LBP) has proven to be an efficient technique for capturing important texture properties. Its simplicity and large success motivated researchers from computer vision and image processing community to study the descriptor and propose variants that overcome identified limitations. One of these limitations is the lack of spatial information in the LBP histograms. Recently, few co-occurrence-based methods have been introduced to overcome this drawback. These techniques improve significantly LBP performance but they are in general much slower than the original operator. In this paper we present two algorithms that make use of the local similarities of the binary patterns to improve the performance of the original LBP without dramatically increasing the execution time. The first algorithm combines the conventional LBP histogram with a histogram recording local similarities of the LBP patterns. The algorithm is almost as fast as the original LBP technique and yet outperforms the operator in terms of retrieval accuracy. The second algorithm records the co-occurrences of LBP codes with Local Similarities of the Patterns. Its retrieval performance is similar to those of the co-occurrence based techniques but with a significant gain in execution time. We have conducted several experiments on two popular datasets (Brodatz, and Outex) to demonstrate the efficiency of the proposed algorithms.**

***Index Terms*—texture descriptor, LBP operator, co-occurrence matrices, local similarities of the patterns**

I. Introduction

Texture is a powerful image feature used by various computer vision and pattern recognition applications. A large number of techniques has been devised for describing and retrieving texture images. The conventional Local Binary Patterns (LBP) has proven to be very successful in capturing important texture properties [1], [2]. This success motivated researchers from computer vision and image processing community to study the descriptor and propose variants that overcome identified limitations. The lack of information about the spatial distribution of the patterns in LBP histograms constitutes a serious drawback of the technique. Recently several co-occurrence-based methods [3]-[5] have been proposed to overcome this drawback. They have proven to be more effective than the conventional LBP technique, but they are significantly much slower. In this paper we introduce two algorithms that capture the local distribution of the LBP patterns by encoding the local similarities of the patterns (LSP). The first algorithm generates a LSP histogram and combines it with the conventional LBP histogram to produce the feature vector characterizing the texture. The second algorithm generates a co-occurrence matrix of the LBP codes and a matrix encoding the local similarities of the LBP patterns. Elements of the co-occurrence matrix constitute the feature vector. Our experiments show that, with a slight increase in the execution time, the first algorithm improves significantly the performance of the conventional LBP technique. They also show that the performance of the second algorithm is similar to that of co-occurrence–based method but requires much shorter execution time. The rest of the paper is organized as follows: section II gives an overview of the LBP operator and some of its variants including those combining co-occurrence matrices and LBP operator. Section III introduces the proposed algorithms and highlights the main differences with similar approaches described in literature, section IV presents the experiments that have been conducted and discusses obtained results. Finally, a conclusion is drawn to summarize obtained results.

II. Brief Review of the Local Binary Patterns Techniques

Local Binary Patterns (LBP) approach was first introduced by Ojala *et al.* in 1996 [1]. The original LBP operator assigns to each image pixel the decimal value of the binary string describing the local pattern around the pixel. Value 1 is assigned to a bit if the grayscale value of the corresponding neighbor is greater than or equal to the grayscale value of the central pixel, otherwise a zero value is assigned to the bit. Fig. 1 illustrates how LBP code is calculated. The LBP code is formally defined as follows (1):

$$LBP(p,r) = \sum_{k=0}^{p-1} S(I_k - I_c) * 2^k \qquad (1)$$

where $S(x) = \begin{cases} 1 \; if \; x \geq 0 \\ 0 \; if \; x < 0 \end{cases}$

where I_k is the kth neighbor of I_c.

Manuscript received December 16, 2015; revised June 20, 2016.

doi: 10.18178/ijsps.4.6.510-514

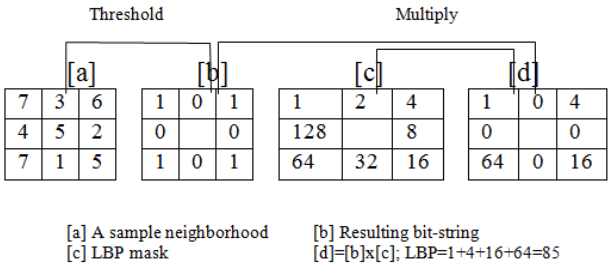

Figure 1. LBP calculation

The LBP operator is characterized by its low computational complexity, resistance to lighting variations and ability to code fine details [6].

Researchers have proposed several variants of the original LBP coding in order to improve its performance and increase its robustness. The number of LBP codes for a neighborhood of P pixels, has been reduced from 2^P to P(P-1)+3, by keeping only LBP codes containing at most 2 transitions (from 1 to 0 or from 0 to 1). These codes are found to constitute the majority of codes in natural images. They are called uniform codes [2]. The following formula (2) is used to calculate the uniformity of an LBP code C_P using the grayscale values in the neighborhood of a central pixel Ic:

$$U(C_P) = |S(I_{P-1} - I_C) - S(I_0 - I_C)| + \sum_{k=1}^{P-1} |S(I_k - I_C) - S(I_{k-1} - I_C)| \quad (2)$$

where $S(x) = \begin{cases} 1 \; if \; x \neq 0 \\ 0 \; if \; x = 0 \end{cases}$

and I_k represents the kth neighbor of the central pixel I_c.

A multi-resolution variant of the LBP, was also proposed [7]. In this approach several LBP codes are assigned to each pixel. For each neighborhood pixels located at distance r from the central pixel, a LBP code is calculated as follows (3):

$$LBP(p,r) = \sum_{k=0}^{p-1} S(I_k - I_c) * 2^k \quad (3)$$

where $S(x) = \begin{cases} 1 \; if \; x \geq 0 \\ 0 \; if \; x < 0 \end{cases}$

And I_k (k=0… p-1) are the grayscale values of the neighboring pixels located at distance r from the central pixel.

Local Ternary Pattern (LTP) [8] and Quinary LBP [6] were introduced in attempt to improve the discriminating power of the original operator by using more than two quantization levels for the intensity difference. Completed LBP (CLBP) [9], combines three types of information to increase the selectivity power of the operator; the description of the texture involves the sign of the local difference which is equivalent to the conventional LBP code, the magnitude of the local differences and the grayscale value of the central pixel. The LBP Variance (LBPV) [10] uses a histogram that accumulates the variance values for each LBP code in attempt to improve the discriminating power of the operator. Median Binary Patterns [11] attempts to increase the robustness of the LBP operator by using the median value in the neighborhood of the central pixel as a threshold for building the LBP code. Centre-Symmetric Binary Pattern (CSLBP) [12] was proposed as a solution to the high dimensionality of the LBP histogram. It allows reducing the feature vector by half. Recently, several researchers [3]-[5] argued that the selectivity power of the LBP operator can be further strengthened by including information about the spatial distribution of the patterns. They proposed using co-occurrence matrices of the LBP codes to capture the spatial information. Actually, the spatial information is not completely absent in the LBP coding, since it records the relation between pixels within a certain vicinity (i.e. sign of the local differences of the grayscale values around each pixel). Unfortunately, the information carried out by LBP histograms is not sufficient to describe relations between complex patterns. Proposed co-occurrence matrices-based techniques go one step further and capture the local distribution of the LBP patterns themselves. The concept of co-occurrence matrix was first introduced on grayscale images I(x,y) by Haralick *et al.* [13]. An element $C_{dx,dy}(u,v)$ of a co-occurrence matrix $C_{dx,dy}$ records the number of co-occurrences of grayscale value u and grayscale value v located at [dx, dy] distance from each other. Formally $C_{dx,dy}$ is defined as follows:

$$C_{dx,dy}(u,v) = \sum_{x=1}^{N} \sum_{y=1}^{M} \delta(u, I(x,y))\, \delta(v, I(x+dx, y+dy))$$

$$\text{where } \delta(x,y) = \begin{cases} 1 \; if \; x = y \\ 0 \; if \; x \neq y \end{cases} \quad (4)$$

N, M= number of rows and columns of image I.

In [3], co-occurrence matrices of 8 displacements are first calculated, then added; elements in the upper triangular part of the resulting matrix and its diagonal are used as feature vector; in order to reduce the dimensionality of the feature vector, only the rotational uniform codes are considered. Unfortunately, the use of rotational uniform codes reduces significantly the retrieval accuracy of the algorithm. In [4], co-occurrence matrices of 4 displacements are first calculated and 6 statistical features (variance, contrast, entropy, energy, inverse difference moment and correlation) are then extracted. A large feature vector consisting of the 4 co-occurrence matrices and the extracted 24 statistical features (6 per matrix) is used as an input to the nearest neighbor classifier for face recognition. In [5], two N4 neighborhoods (4 adjacent pixels) are considered when calculating the LBP codes, which reduces the number of LBP codes to only 16 (instead of the original 256) for each neighborhood. Co-occurrence matrices of the obtained LBP codes are then calculated and used as feature vectors. A nearest neighbor classifier is then applied to the resulting vector and used for face recognition and texture classification.

III. Proposed Algorithms

The algorithms proposed in [3]-[5] extract several co-occurrence matrices of the LBP image. The feature vector is either made of all or some elements of those matrices or consists of a number of statistical values extracted from those matrices. Calculation of a co-occurrence matrix is relatively expensive and the use of all or most of its elements leads to a high dimensionality feature vector which slows down the similarity measurement. In this paper we describe two efficient techniques.

A. LS_LBP Technique

The first technique, called Local Similarities of the LBP patterns (LS_LBP), combines two histograms: the traditional LBP histogram and a histogram that records the number of different LS_LBP patterns. An LS_LBP pattern describes existing similarities between the uniform LBP code of a pixel and the uniform LBP codes of its neighbors. By doing so, information about the local arrangements of the LBP patterns is captured and added to the already efficient LBP histogram. Local similarities information is described using a coding similar to the one adopted by the conventional LBP:

$$LS\_LBP\_(p,r) = \sum_{k=0}^{p-1} f(LBPI_k, LBPI_c) * 2^k \qquad (5)$$

where $f(x,y) = \begin{cases} 1 \; if \; x = y \\ 0 \; if \; x \neq y \end{cases}$

$LBPI_c$ is the central pixel in the LBP image

$LBPI_k$ is the kth neighbor of $LBPI_c$

We would like to indicate that this technique is completely different from the one proposed in [14] and called Local Similarities Patterns (LSP), which is actually not much different from the LTP technique introduced in [8].

LS_LBP Algorithm

1. Convert input image I into a grayscale image G.
2. Calculate the LBP Image (LBPI); each pixel value in LBPI represents the uniform LBP code of the corresponding pixels in the original image G. In this work, a 3x3 neighborhood is used.
3. Calculate the LS_LBP image (LS_LBPI) that represents the local similarities of the LBP codes within the selected neighborhood using equation (5). In this work, a 3x3 neighborhood is used.
4. Calculate, from the LBPI image, the LBP histogram (LBPH).
5. Calculate from LS_LBPI image, the LS_LBP histogram (LS_LBPH)
6. Feature_V= [LBPH, LS_LBPH]

B. LBP_LSP_CO Technique

The second algorithm is introduced to further improve the retrieval accuracy of the LS_LBP algorithm at the cost of a reasonable increase in the execution time. We have noticed that co-occurrence–based techniques improve significantly the retrieval accuracy of the conventional LBP method. But they are in general very slow compared to the LBP method. The proposed algorithm consists of recording the co-occurrence of the uniform LBP patterns and the LS_LBP patterns. In order to reduce the dimensionality of the feature vector and therefore reduce the execution time, we consider two 4-neighbor matrices (LS_LBP1 and LS_LBP2) instead of the 8-Neighbor previously utilized (see Fig. 2). This will reduce the possible values of the LS_LBP codes from 256 ([0-2^8-1]) to 16 ([0, 2^4-1]). The two matrices will be then combined into a single matrix by using a bitwise-or of their elements. The elements of the new matrix are still in the range of 0-15 and yet they carry information extracted from both matrices LS_LBP1 and LS_LBP2. A co-occurrence matrix of the uniform LBP codes and the combined LS_LBP codes is used as a feature vector describing the texture.

LBP1	LBP2	LBP3
LBP8	LBPc	LBP4
LBP7	LBP6	LBP5

a)- Original 8-Neighborhood used to generate LS_LBP image

	LBP1	
LBP4	LBPc	LBP2
	LBP3	

LBP1		LBP2
	LBPc	
LBP4		LBP3

b)- The two 4-neighborhoods used to generate LS_LBP1 and LS_LBP2

Figure 2. Splitting 8-neighborhood into 2 4-neighborhoods

Calculation of LS_LBP1 and LS_LBP2 elements is performed using equation (5) where p=4 instead of p=8. Calculation of the LBP_LSP co-occurrence matrix uses the formula (6) shown below:

$$LBP\_LSP\_CO(u,v) = \sum_{x=1}^{N}\sum_{y=1}^{M} \delta(u, LBPI(x,y))\, \delta\big(v, LS\_LBP(x,y)\big)$$

$$\text{where } \delta(x,y) = \begin{cases} 1 \; if \; x = y \\ 0 \; if \; x \neq y \end{cases} \qquad (6)$$

N, M= num. of rows and colus of LBPI and LSP_LBP

LBP_LSP_CO Algorithm

1. Convert input image I into a grayscale image G.
2. Calculate the LBP image (LBPI); each pixel value in LBPI represents the uniform LBP code of the corresponding pixels in the original image G. In this work, a 3x3 neighborhood is used.
3. Calculate the two local similarity matrices LS_LBP1 and LS_LBP2 using equation (5) and P=4.
4. Combine the two matrices using bitwise-or operator LS_LBP=LS_LBP1 .OR. LS_LBP2
5. Calculate the LBP LS_LBP co-occurrence matrix (LBP_LSP_CO) using equation (6).
6. Feature_V= [LBP_LSP_CO(:)]

C. Similarity Measurement

Several distance metrics are available in literature. In this work we use the well-known Chi2 distance; this choice is motivated by the fact that it's widely adopted by LBP-based methods.

Given two texture images I and J, represented by two N-dimensional feature vectors F^I and F^J; The Chi2 distance function between I and J is defined by the following formula:

$$Chi2(F^I, F^J) = \frac{1}{2}\sum_{k=1}^{k=N} \frac{({F^I}_k - {F^J}_k)^2}{{F^I}_k + {F^J}_k} \quad (7)$$

IV. Experiments and Discussion

A. Test Datasets

Texture images that are used in the experiments are selected from 2 popular datasets: Brodatz album [15], and Outex database from Oulu University [16]. Selected images are those that have roughly uniform texture (i.e. similar texture over the whole image). From Brodatz database, 80 images of size 640 x 640 pixels are used; from each of these images 16 non-overlapping sub-images of size 128x128 are extracted, which gives a database of 1216 images. From outex-TR_00000 (texture retrieval test suit), 86 different uniform textures have been selected, from which 1376 (86x16) sub-images of size 128x128 are extracted.

B. Hardware and Software Environment

Our experiments have been conducted on an Intel Core i5 (2.5GHz) laptop with 4GB RAM. The software environment consists of MS Windows 7 professional and MatlabR2013.

C. Performance Evaluation

The performance evaluation of the proposed approach is based on two indicators: the well-known precision/recall values, initially introduced by Kankahalli *et al*. [17] and the average execution time. The average execution time consists of the average time for extracting the feature vector and the average time for calculating the similarity between the feature vectors of the query image and the database image. The first time depends on the complexity of the feature extraction algorithm and the second one depends on the size of the feature vector.

The average Precision Value (PV) is calculated for each dataset according to the following formulas:

For each query image, retrieved images are sorted in descending order then its average precision Pq for the first N (in this work N=16) retrieved images is calculated as follows:

$$Pq = \frac{1}{N}\sum_{i=1}^{N} \delta(Ii, Iq) \quad (8)$$

$$\delta(x,y) = \begin{cases} 1 & if\ x\ and\ y\ extracted\ from\ the\ same\ image \\ 0 & if\ x\ and\ y\ are\ extracted\ from\ different\ images \end{cases}$$

Then the average precision for all query images of that dataset is calculated as follows:

$$PV = \frac{1}{|QDB|}\sum_{q=1}^{|QDB|} Pq \quad (9)$$

where |QDB| is the size of the query database (i.e. number of queries).

We have also included Precision-Recall curves to visually evaluate the difference in the performance of the techniques.

D. Experiments

In our experiments, the query dataset (QDB) is the whole dataset; therefore all images of a dataset have been used in turn as a query. The average precision (for the first 16 retrieved images) and the precision-recall graph are calculated. Therefore 1216 queries have been made for Brodatz dataset and 1376 queries for Outex. Beside the proposed techniques LS_LBP and LBP_LSP_CO, two other techniques have been evaluated in these experiments. The conventional LBP [1], and the LBP_CO [3]. In LBP_CO we used uniform patterns instead of the rotational invariant uniform patterns used in LBP_CO, because we found that the latter operator performance is significantly lower than the former when applied to our datasets.

E. Results and Discussion

Fig. 3 and Fig. 4 show the precision recall graphs of the 4 techniques included in this experiment obtained on Brodatz and Outex datasets respectively.

Table I and Table II summarize the results obtained on the two datasets by different techniques. Each table shows the average Precision Value (PV), extraction time (Ext-time), and comparison times (Comp-time) required by each technique.

TABLE I. Comparing the Performance of the Tested Methods on the Brodatz Dataset

	Brodatz Dataset			
	LBP	**LBP_CO**	**LS_LBP**	**LBP_LSP_CO**
PV(%)	89.52	92.03	90.83	91.56
Ext-time(s)	0.003	0.007	0.003	0.003
Comp-time(s)	0.003	0.013	0.005	0.007

TABLE II. Comparing the Performance of the Tested Methods on the Outex Dataset

	Outex Dataset			
	LBP	**LBP_CO**	**LS_LBP**	**LBP_LSP_CO**
PV(%)	84.43	89.69	87.90	89.17
Ext-time(s)	0.003	0.007	0.003	0.003
Comp-time(s)	0.004	0.013	0.006	0.008

The first observation we can make is that LS_LBP improves significantly the performance of the conventional LBP (from 89.52 to 90.83 for Brodatz and from 84.43 to 87.90% for Outex) with a slight increase in the execution time (0.006 for LBP and 0.008 for LS_LBP). The increase in time is mainly due to the increase in the feature vector size from 59 to 512 which makes the similarity calculation a little bit slower.

The second observation we can make is that the co-occurrence information improves consistently the discriminating power of the techniques (compare performance of LBP and LS_LBP with LBP_CO and LBP_LSP_CO).

Finally, we can see that the performance of the proposed LBP_LSP_CO is similar to the one of LBP_CO (91.55% against 92.02% for Brodatz and 89.17% against

89.69% for Outex) but requires only half of the execution time (0.010s against 0.020 for Brodatz and 0.011 against 0.020 for Outex).

The precision-recall graphs shown in Fig. 3 and Fig. 4 illustrate the improvement produced by the proposed descriptors.

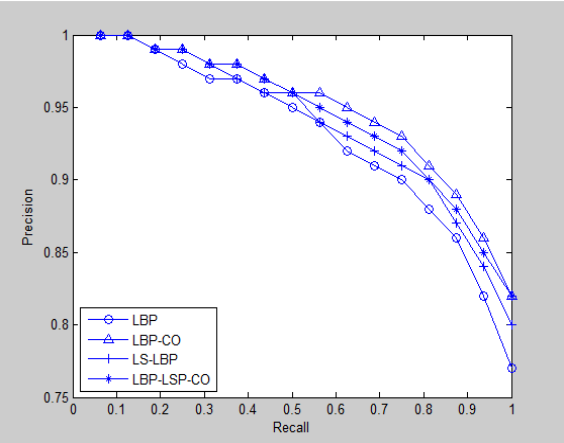

Figure 3. Precision recall graph obtained on Brodatz dataset.

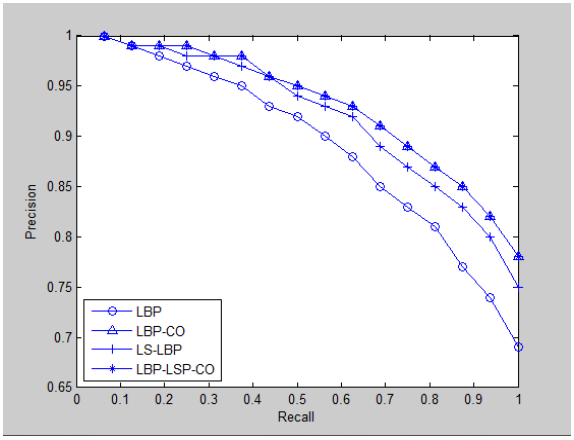

Figure 4. Precision recall graph obtained on Outex dataset.

V. Conclusion

This paper describes two texture retrieval descriptors that take into account some information about the spatial arrangement of the LBP patterns (local similarities of the LBP patterns). The first descriptor, called Local Similarities of the LBP patterns (LS_LBP), combines the traditional LBP histogram and a histogram that records the number of different LS_LBP patterns. An LS_LBP pattern describes existing similarities between the uniform LBP code of a pixel and the uniform LBP codes of its neighbors. By doing so, information about the local arrangements of the LBP patterns is captured and added to the already efficient LBP histogram. The resulting algorithm produces a better retrieval accuracy than that of the conventional LBP with a slight increase in the execution time. The second descriptor records the co-occurrence of the uniform LBP patterns and the LS_LBP patterns. This addition improves significantly the performance of the LS_LBP algorithm; its performance is similar to that of the descriptor using co-occurrence matrix, but it requires only half of the time because of the reduction in LS_LBP codes involved in the co-occurrence matrix.

References

[1] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, pp. 51-59, 1996.

[2] T. Ojala, M. Pietikäinen, and T Mäenpää, "Gray scale and rotation invariant texture classification with local binary patterns," in *Proc. 6th European Conference on Computer Vision*, 2000.

[3] N. Shadkam and M. S. Helfroush, "Texture classification by using co-occurrences of local binary patterns," in *Proc. 20th Iranian Conference on Electrical Engineering* Tehran, Iran, May 15-17, 2012, pp. 1442-1446.

[4] A. H. Bishak, Z. Ghandriz, and T. Taheri, "Face recognition using Co-occurrence Matrix of Local Average Binary Pattern (CMLABP)," *Journal of Selected Areas in Telecommunication*, vol. 2, pp. 15-19, 2012.

[5] R. Nosaka, Y. Ohkawa, and K. Fukui, "Feature extraction based on co-occurrence of adjacent local binary patterns," *LNCS*, vol. 7088, pp. 82-91, 2011.

[6] L. Nanni, A. Lumini, and S. Brahnam, "Survey on LBP based texture descriptors for image classification," *Expert System with Applications*, vol. 39, pp. 3634-3641, 2012.

[7] T. Ojala, M. Pietikaeinen, and T. Maeenpaea, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971-987, 2002.

[8] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *LNCS*, vol. 4778, pp. 168-182, 2007.

[9] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Trans. on Image Processing*, vol. 19, pp. 1657-1663, 2010.

[10] Z. Guo, L. Zhang, and D. Zhang, "Rotation invariant texture classification using LPB Variance (LBPV) with global matching," *Pattern Recognition*, vol. 43, pp. 706-719, 2010.

[11] A. Hafiane, G. Seetharaman, K. Palaniappan, and B. Zavidovique, "Rotationally invariant hashing of mediane binary pattern for texture classification," in *Proc. ICIAR*, 2008, pp. 619-629.

[12] M. Heikkila, M. M. Pietikainen, and C. Schmid, "Descroiption of interest regions with local binary patterns," *Pattern Recognition*, vol. 42, pp. 425-436, 2009.

[13] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Texture features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, pp. 610-621, 1973.

[14] M. Masoudifar, H. PourReza, and M. ManafZadeh, "LSP: Local similarity pattern, a new approach for rotation invariant noisy texture analysis," in *Proc. 18th International Conference on Image Processing*, Brussels, Belgium, September 11-14, 2011, pp. 837-840.

[15] Brodatz textures. [Online]. Available: http://www.ux.uis.no/~tranden/brodatz.html

[16] Outex is a framework for empirical evaluation of texture classification and segmentation algorithms. [Online]. Available: http://www.outex.oulu.fi

[17] M. Kankahalli, B. M. Mehtre, and J. K. Wu, "Cluster-Based color matching for image retrieval," *Pattern Recognition*, vol. 29, pp. 701-708, 1996.

Dr. Abdelhamid Abdesselam obtained his PhD in Computer Vision from Ecole National Polytechnique de Touloue, France in 1992. He joined University Malaysia Sarawak in November 1994, Qatar University in April 2000, currently he is working with Sultan Qaboos University as Asst. Professor of Computer Science. His research interests include content based image retrieval, biometrics, medical image analysis and machine learning.