Greedy Mean Squared Residue for Texture Images Retrieval

Salah Bougueroua and Bachir Boucheham

Abstract In this paper, we propose a new algorithm for texture retrieval, using clustering strategy. Indeed, it is largely noticed that in existing CBIR systems and methods, the collection of the images similar to the query is realized on the basis of comparison of the database images to the query solely. Hence, the results might not be globally homogeneous. In this paper, the collection of the images most similar to the query is realized considering the global homogeneity of the whole cluster (result). Knowing that this is of an exponential order problem, we use a greedy solution consisting in growing the cluster corresponding to a query, one image at a time, based on the *Mean Squared residue* measure of Cheng and Church (Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, 2000) [1], originally proposed for the biclustering of gene expression data. At each stage, the new added image to the cluster will be that that preserves most the homogeneity of the current cluster. The texture descriptor used in this work is the uniform-LBP. Experimentations were conducted on two texture image databases, Outext and Brodatz. The proposed algorithm shows an interesting performance compared to the uniform-LBP combined to Euclidean metric.

Keywords CBIR • Image retrieval • Biclustering • Mean squared residue • Texture • Similarity measure • Greedy search • Optimization

---

S. Bougueroua (✉) · B. Boucheham
Department of Computer Science, 20 Août 1955 University of Skikda,
B.P. 26 Route d'El-Hadaiek, 21000 Skikda, Algeria
e-mail: s_bougueroua@yahoo.fr; s.bougueroua@univ-skikda.dz

B. Boucheham
e-mail: boucheham_bachir@yahoo.fr; b.boucheham@univ-skikda.dz

S. Chikhi et al. (eds.), *Modelling and Implementation of Complex Systems*, Lecture Notes in Networks and Systems 1,
DOI 10.1007/978-3-319-33410-3_9

1 Introduction

Content Based Image Retrieval (CBIR), [1] is an alternative technique which has been proposed in the early 1980s [2] to overcome the disadvantages of the text based retrieval (TBIR). These disadvantages are summarized into two aspects; hardness of the annotation, especially for large databases, and subjectivity of the attributed keywords.

CBIR systems are essentially composed of two steps: *features extraction* from low level visual attributes, and *the retrieval of visually similar images* using one of the similarity measures. The most exploited visual attributes in the literature are Color, Texture and Shape. Specifically, texture is a characteristic that is ubiquitous in objects in general, remote sensing and scenes images particularly. This puts texture as privileged feature that captures attention of researchers in the computer vision fields (classification, retrieval, recognition, etc.).

In the context of image characterization based on texture, the "co-occurrence matrices" of Haralick et al. [3] is one of the earliest methods. This was specifically proposed for texture classification. The Local Binary Pattern (LBP) of Ojala et al. [4, 5] is another technique for texture feature extraction. Although simple, this method is very effective technique. Particularly, the LBP operator has known great success since its apparition in texture retrieval applications. It was also successfully used in face recognition [6], gender and age classification [7]. For example, the combination LBP-GLCM (Gray Level Co-Occurrence Matrix) showed interesting results for tea leaves classification [8]. Due to its success, a significant number of LBP variants were introduced, such as ILBP [9], LTP [10], DLBP [11], block-based LBP [12], ElLBP [13], GLIBP [14].

Despite the success of the above mentioned methods in image retrieval by content, these methods show generally difficulties to stay effective in all situations. For this reason, usage of more sophisticated methods has been imposed as a necessity, such as, the combination of different visual attributes, e.g. [15–22].

In image retrieval, the comparison step plays a crucial role. In this context, many measures were proposed in the literatures, such as: *histograms intersection* of Swain and Ballard [23], the *Earth Mover's Distance* (EMD) [24], and *Comparing Histograms by Clustering* (CHIC) [25]. Other more simple metrics are also widely used, such as the *Euclidean*, *Manhattan* and *Canberra* metrics. Some of the above metrics and other ones are compared by the authors of [26, 27]. Their results show clearly the important role that the distance metrics play.

The present work proposes a contribution in the comparison step of CBIR systems. Unlike the strategy of classical approaches in image retrieval methods by content, that consists in selecting the resulting images matching the query, one at time, but, by ignoring, in each step, the previous selected images; our contribution is based on the assumption that: *the feature vectors of the images which are visually similar have the same variation (co-regulation) over the features*. Following this assumption, we focused our attention on (Bi)-Clustering techniques. Particularly, the *Mean Squared Residue* (MSR) is a measure introduced by Cheng and Church [1] for the biclustering of gene expressions. To the best of our knowledge, no previous works in the CBIR field have used this measure. Indeed, the proposed algorithm is distinguished by the fact that all images in the current set of selected images contribute in the selection of the next image, at each step.

The remaining of this paper is organized as follows. In the next section, we address briefly materials related to our work; including biclustering and the Mean Squared Residue (MSR). Our proposal is presented in the Sect. 3. The experimentations and obtained results are discussed in the Sects. 4 and 5, respectively. A conclusion and perspectives are expressed in the Sect. 6.

2 Materials and Methods

2.1 The Biclustering

By contrast to the clustering techniques, which classify the samples over all the features (values), the biclustering techniques aim to cluster simultaneously the rows and the columns. In expression data analysis, the rows correspond to the genes and the columns correspond to the conditions. Accordingly, the biclustering of gene expression data aims to find the subset(s) of genes which they are highly correlated over a subset(s) of conditions.

Formally, let A be the expression data composed of the set of the genes (rows) X and the set of conditions (columns) Y.

The biclustering techniques try to find the bicluster (sub-matrix) A_{IJ}, where $I \subseteq X$ and $J \subseteq Y$ such that the genes in I, are highly co-regulated over the set of conditions J.

2.2 The Mean Squared Residue (MSR)

Cheng and Church [1] introduced an efficient node-deletion algorithm for the biclustering of the expression data. Their algorithm attempts to find the maximum bicluster with low Mean Squared Residue.

The MSR *H* of the bicluster with the set of rows and columns I and J, respectively, is calculated as follow:

$$H(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2 \tag{1}$$

where

$$a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij} \tag{2}$$

$$a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij} \tag{3}$$

And

$$a_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} a_{ij} \tag{4}$$

More information on biclustering techniques can be found in the interesting survey of Madeira and Oliveira [28].

3 The Mean Squared Residue for Texture Retrieval

Our proposed method is based on the MSR measure, with adaptation to the CBIR case. The matrix (expression data) is replaced by the index (i.e. database images feature vectors), where the rows correspond to the images and the columns correspond to the features.

Since we are interested by the image retrieval based on texture, we have chosen the LBP method for texture features extraction. Indeed, the LBP operator and its various variants are known for their effectiveness and efficiency. However, the use of the LBP histogram is not the only descriptor allowed in our algorithm and it can be generalized to other descriptors, such as, color histogram.

The LBP histogram of an image is 256-valued (when 8 neighbors are considered). Therefore, for more compactness, we have used the LBP uniform ($LBP^{u2}_{P,R}$) [4], radius (R) equal to 1, and 8 neighbors (i.e. P = 8). This yields 58 uniform patterns (i.e. histogram of 59 bins, one bin for the non uniform pattern). Consequently, if the database is composed of M images, the matrix will be of $M \times 59$ values.

The proposed algorithm is composed of two main steps, as follows.

Step 1 *Initialization*: Let I be the cluster representing the selected images corresponding to a given query Q. In this step, the cluster is initiated to *empty*. (The first step in Fig. 1)

Step 2 *Growing*: in this step, I is made grow by adding to it selected images from the database, one at a time, so that the currently added image is that that preserves most the overall homogeneity of I. For this purpose, we used a

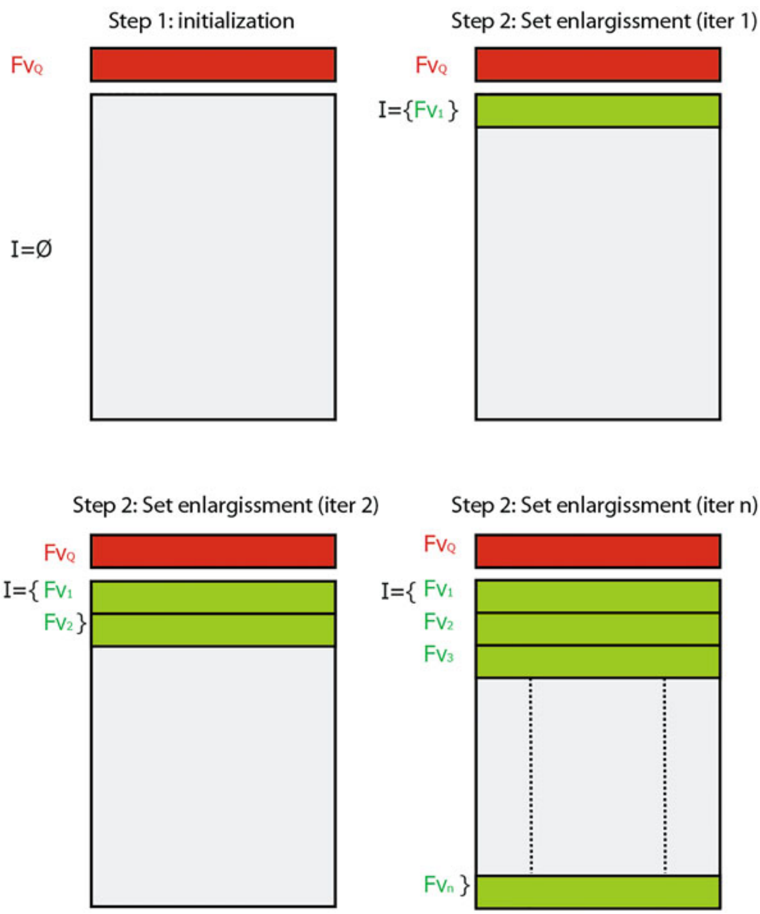

Fig. 1 Steps of growing set *I*

greedy search, where the image which causes the lowest *cost* is included to the cluster (Eq. 5). By "cost", we mean the *difference* between the MSR of the cluster *before* and the MSR *after adding* the feature vector of image i (Eq. 6). This step is repeated until satisfying a stop criterion. The criterion we used is the number of images in the cluster (i.e. the number of cluster rows)

$$\Delta MSR_i(Q, I) = H(Q \cup I, Y) - H(Q \cup I \cup \{i\}, Y) \tag{5}$$

$$\arg\min_i(\Delta MSR_i(Q, I)) \tag{6}$$

A description of the proposed algorithm is given below:

```
Algorithm: texture image retrieval using the MSR;
Input: index:matrix[m,n];
//m the number images in database.
//n the number of patterns
Q:vector; // Query Feature vector
sizeCR:integer; // the maximum number of rows (images) in
//the result cluster I.
Output: the set I of images which are visually most
//similar to the query Q.
Begin
I=Ø;
While (size(I)<sizeCR) do
  begin
  //Search for image i which brings the minimum ΔMSR
  //among the images not included in the current cluster.
  i = argmin(ΔMSR_i(Q,I))
  I=I∪{ Fv_i };
  End;
End.
```

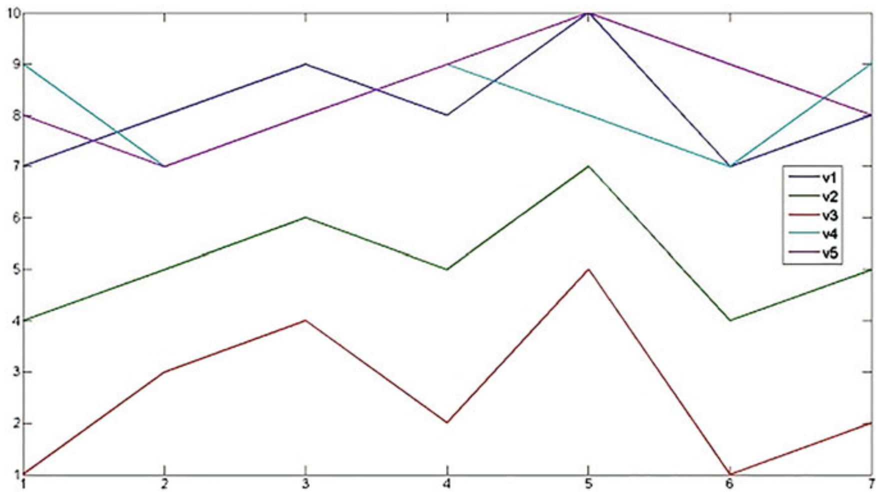

Fig. 2 Example, the five vectors

For example, let the following five vectors (Fig. 2): V_1, V_2, V_3, V_4, and V_5 such that:

$V_1 = [7, 8, 9, 8, 10, 7, 8]$.
$V_2 = [4, 5, 6, 5, 7, 4, 5]$.
$V_3 = [1, 3, 4, 2, 5, 1, 2]$.
$V_4 = [9, 7, 8, 9, 8, 7, 9]$.
$V_5 = [8, 7, 8, 9, 10, 9, 8]$.

We also assume that we have a query vector V_Q, such that:

$V_Q = [7, 8, 9, 10, 7, 8]$.

If we match the vector V_Q with the above five vectors using a classical distance metric such as Euclidean distance (Eq. 7) they will be ranked according to their distances from V_Q as follow V_1 (d = 0), V_5 (d = 2.83), V_4 (d = 3.46), V_2 (d = 7.94), and V_3 (d = 14.8).

Now, using the proposed method based on the MSR. At the first step (initialization) we have only the query vector V_Q. In the second step, we search for the vector among the database vectors (not included yet) that has the lowest cost (MSR difference (ΔMSR)). So, according to the vectors that we have, V_1 is the one which verifies this constraint ($\Delta MSR_{V1} = 0$). We repeat the process, we have $Fv_Q = V_Q$ and $I = \{V_1\}$, the vector which has the lowest cost when it is included to the set I is V_2 ($\Delta MSRv_2 = 0$), then V_3 ($\Delta MSR_{V3} = 0.0459$), and after that V_5 ($\Delta MSRv_5 = 0.1957$). In this step $Fv_Q = V_Q$ and $I = \{V_1, V_2, V_3, V_5\}$ and the inclusion of V_4 has the cost ($\Delta MSR_{V4} = 0.1926$). At the end, the resulted vectors will be ranked as follow: V_1, V_2, V_3, V_5, and V_4.

$$d_{euc}(x, y) = \sqrt{\sum_i (x_i - y_i)^2} \tag{7}$$

4 Experimentations

Our experimentations were conducted on two databases of textured images. The first one is Outex_tc_00000 test suite of *Outex*[1] database. This suite is composed of 480 images, distributed on 24 classes, 20 images for each class, with a size of 128 × 128 pixels. Some samples from this database are shown in Fig. 3.

The second database is *Brodatz*[2] database. This dataset is composed of 112 textured images, having 640 × 640 pixels size. In this paper, we have divided each image into 16 non-overlapping sub-images (4 × 4). Thus, the total number of images is 1792 images, 16 images in each class. Figure 4 presents some samples from this database.

[1]http://www.outex.oulu.fi/db/classification/tmp/Outex_TC_00000.tar.gz.

[2]http://multibandtexture.recherche.usherbrooke.ca/images/Original_Brodatz.zip.

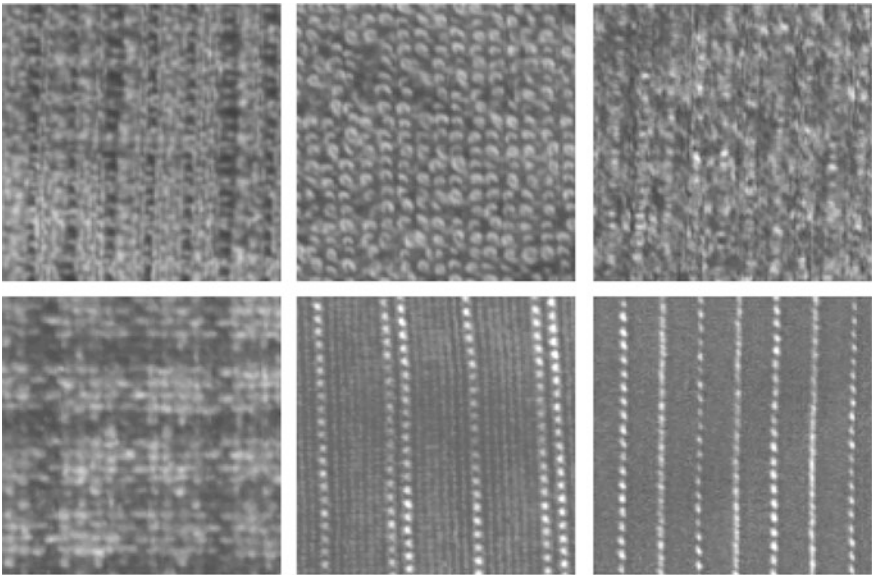

Fig. 3 Some samples from Outex database (128 × 128 px)

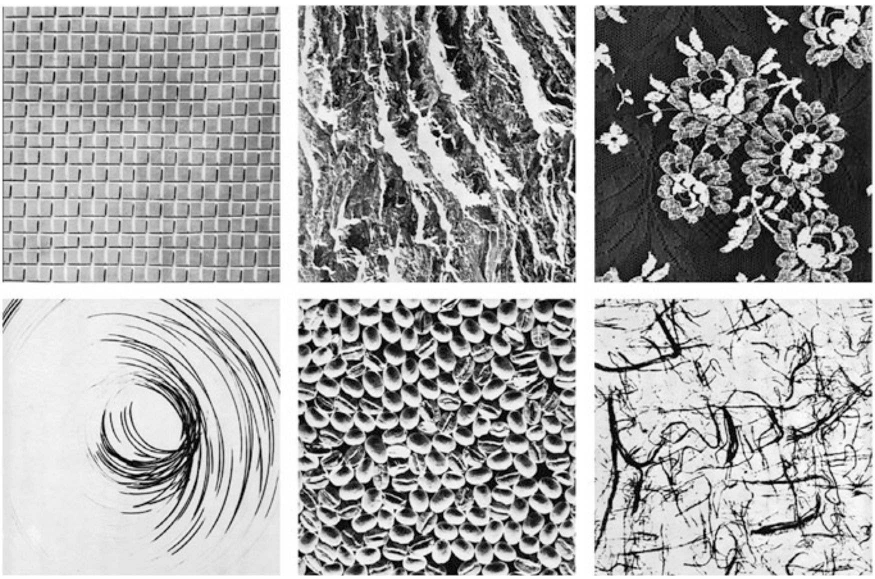

Fig. 4 Some samples from Brodatz database (640 × 640 px)

The performance evaluation is done through the Average Precision (AP) criterion. The AP of texture class I, which contains m samples, within a window of k images (the number of retrieved images), is calculated as follows:

$$AP(I,k) = \frac{\sum_{i}^{m} p(i,k)}{m} \quad (8)$$

$$p(i,k) = \frac{\text{Nbr Relevant Retrieved Images}}{\text{k}} \quad (9)$$

5 Results and Discussion

Table 1 presents the results of the retrieval using the proposed algorithm based on the MSR versus Euclidean distance metric in term of the average precision of the top 20 images retrieved (i.e. k = 20), over all database. The table shows clearly that the MSR is more effective than Euclidean metric, where an AP of 87.68 % overall the database is reached against 85.18 % for Euclidean distance. More of that, the results analysis by class show that when Euclidean reaches 90 % of AP, the MSR gives a perfect AP (100 %) (With the exception, the class C3).

The second table (Table 2) presenting the AP for the second database (Brodatz) of the top 16 retrieved images, also shows an interesting performance for the

Table 1 Outex_tc_00000 database (precision, window = 20)

Texture	Proposed	Euclidean	Texture	Proposed	Euclidean
C0	**100**	96.5	C12	**100**	95.5
C1	**100**	96	C13	50	**50.3**
C2	**100**	90.5	C14	40	**50.5**
C3	**99.5**	90.5	C15	**74**	67.5
C4	100	100	C16	56.3	**57**
C5	100	100	C17	**58**	55.3
C6	100	100	C18	**100**	97.3
C7	100	100	C19	**78.8**	74.5
C8	**100**	98.3	C20	**100**	96
C9	**77.8**	68.3	C21	**100**	99.8
C10	100	100	C22	**95**	89
C11	**100**	99	C23	**75**	72.5
			Total	**87.68**	85.18

Table 2 Brodatz database (16 sub-images * 112)

Texture	Proposed	Euclidean	Texture	Proposed	Euclidean	Texture	Proposed	Euclidean
D1	**94.5**	89.8	D39	25	**32.8**	D77	100	100
D2	43.4	**44.1**	D40	**99.2**	86.7	D78	**100**	85.2
D3	63.7	**72.7**	D41	**93.8**	85.2	D79	**94.1**	85.5
D4	**74.2**	73	D42	58.6	**66.4**	D80	**100**	89.1
D5	45.3	**49.6**	D43	13.3	**17.6**	D81	**100**	94.9
D6	100	100	D44	19.1	**21.9**	D82	**100**	95.3
D7	43	**46.1**	D45	14.1	**23**	D83	100	100
D8	39.8	**41.4**	D46	**94.1**	82.4	D84	**100**	99.2
D9	**59.4**	55.9	D47	**100**	96.5	D85	**100**	99.2
D10	**94.5**	85.2	D48	**71.1**	62.9	D86	**100**	98.8
D11	60.2	**63.3**	D49	100	100	D87	94.1	**94.9**
D12	**89.5**	78.9	D50	**94.5**	80.9	D88	26.2	**40.6**
D13	25	**36.7**	D51	**83.6**	81.6	D89	39.8	**49.6**
D14	94.1	**94.9**	D52	**100**	99.2	D90	48.8	**50**
D15	**58.2**	56.6	D53	100	100	D91	33.2	**40.2**
D16	100	100	D54	44.5	**48.4**	D92	**98**	94.1
D17	**100**	94.5	D55	100	100	D93	50	**66**
D18	**100**	89.5	D56	100	100	D94	61.3	**62.9**
D19	67.6	**69.9**	D57	100	100	D95	**89.1**	85.2
D20	100	100	D58	28.9	**33.2**	D96	**87.9**	84
D21	100	100	D59	14.8	**23.4**	D97	50.4	50.4
D22	**53.9**	52.7	D60	**72.7**	66.4	D98	**61.7**	57
D23	33.2	**41.4**	D61	**69.1**	63.3	D99	36.7	**41**
D24	**82.8**	72.3	D62	**65.6**	63.7	D100	36.7	**39.5**
D25	63.3	**66.4**	D63	31.3	**35.5**	D101	**100**	97.3
D26	**87.5**	80.9	D64	100	100	D102	**100**	98.8
D27	**50**	49.2	D65	100	100	D103	**100**	97.3
D28	**94.1**	85.2	D66	**94.9**	89.5	D104	**94.1**	87.1
D29	**100**	92.2	D67	**88.7**	80.5	D105	76.6	**79.7**
D30	41	**48**	D68	**97.7**	96.1	D106	60.5	**70.3**
D31	29.3	**36.3**	D69	48	**49.6**	D107	47.3	**53.1**
D32	100	100	D70	**100**	97.3	D108	**68**	67.2
D33	**100**	98.8	D71	**71.1**	68.8	D109	**99.2**	91.8
D34	**98**	86.7	D72	30.1	**38.7**	D110	**100**	99.2
D35	28.9	**33.2**	D73	32.8	**41.4**	D111	**100**	79.7
D36	29.7	**41**	D74	**93**	77.3	D112	**83.2**	74.6
D37	**82**	80.9	D75	**92.6**	86.7			
D38	36.7	**42.2**	D76	**100**	93.4	**Total**	**73.59**	72.67

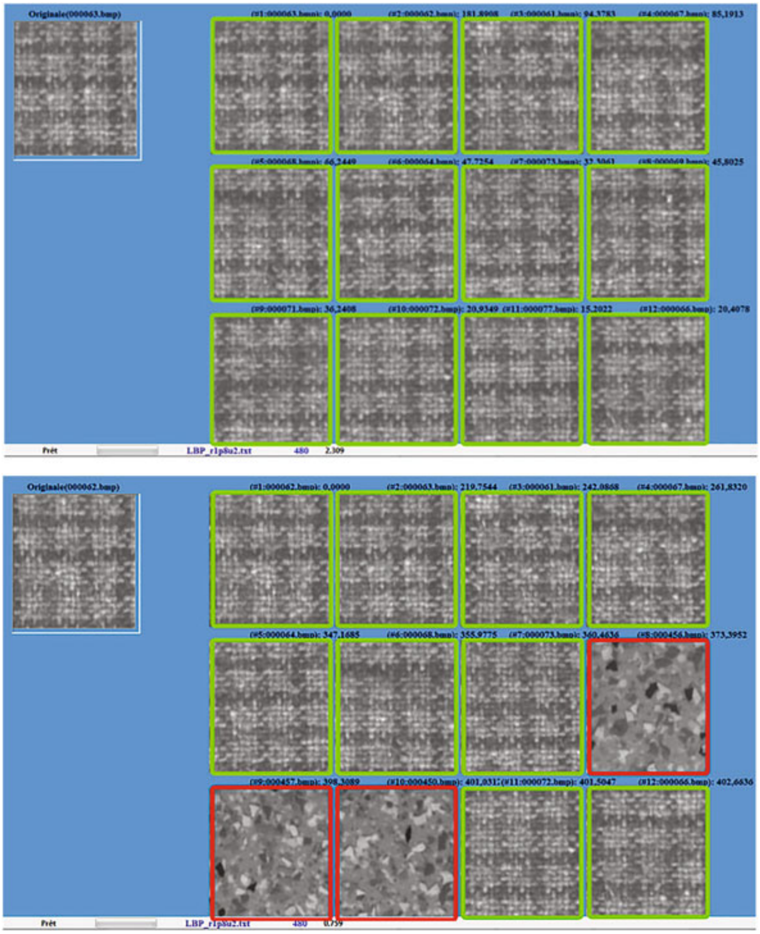

Fig. 5 Results obtained by querying with image #000062; *top* proposed method (precision = 12/12), *bottom* retrieval using Euclidean distance (precision = 9/12)

proposed method against Euclidean distance. Actually, the proposed method beats the Euclidean distance 57 times, and loses 40 times, and shows comparable performance to the Euclidean distance in 15 classes.

The screenshots of the results obtained using the proposed method and Euclidean distance on two different queries are shown in Figs. 5 and 6.

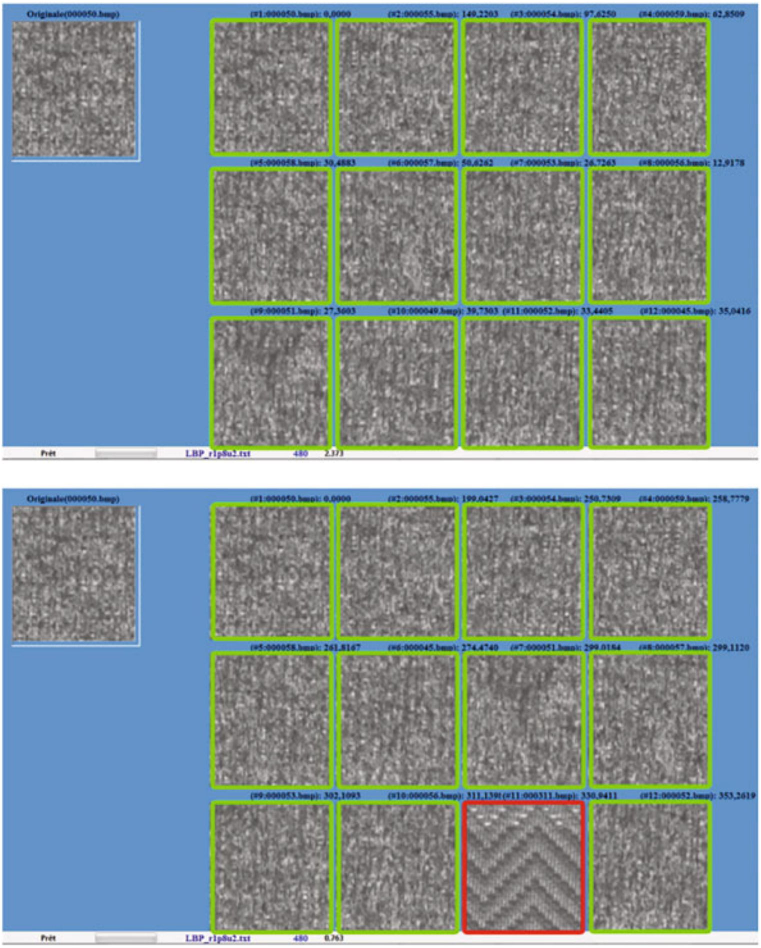

Fig. 6 Results obtained by querying with image #000050; *top* proposed method (precision = 12/12), *bottom* retrieval using Euclidean distance (precision = 11/12)

6 Conclusion

In this paper, we have proposed a new algorithm for image retrieval based on texture. Specifically, we have used the uniform-LBP for extracting texture features. The proposed algorithm is based on the Mean Squared Residue measure, an existing measure originally proposed for biclustering of gene expressions data. Our algorithm starts by initialization of the result set to empty. Then, in each iteration, a new image is selected from the database images and added to the result set, so that the overall homogeneity of the result set is preserved most. The algorithm is distinguished by the fact that all images in the result set contribute to the selection of the new image through the difference of the MSR measures.

The obtained results of the conducted experimentation on Outex and Brodatz databases show an interesting performance (Average Precision) compared to the uniform-LBP combined to Euclidean distance.

In the future, we plan investigation of the proposed algorithm performance using other descriptors, such as color histograms, and also combination with other standard metrics (Canberra, Manhattan, etc.). Furthermore, the adaptation of the proposed algorithm for features selection is also envisaged.

References

1. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (2000)
2. Liu, Y., Zhang, D., Lu, G., Ma, W.-Y.: A survey of content-based image retrieval with high-level semantics. Pattern Recogn. **40**, 262–282 (2007)
3. Haralick, R.M., Shanmugam, K., Dinstein, I.H.: Textures features for image classification. IEEE Trans. Syst. Man Cybern. **SMC-3**, 610–621 (1973)
4. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. **24**, 971–987 (2002)
5. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Pattern Recogn. **29**, 51–59 (1996)
6. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: application to face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **28**, 2037–2041 (2006)
7. Yang, Z., Ai, H.: Demographic classification with local binary patterns. In: Lee, S.-W., Li, S. (eds.) Advances in Biometrics, vol. 4642, pp. 464–473. Springer, Berlin (2007)
8. Tang, Z., Su, Y., Er, M.J., Qi, F., Zhang, L., Zhou, J.: A local binary pattern based texture descriptors for classification of tea leaves. Neurocomputing **168**, 1011–1023 (2015)
9. Jin, H., Liu, Q., Lu, H., Tong, X.: Face detection using improved LBP under Bayesian framework. In: Proceedings of the Third International Conference on Image and Graphics (ICIG'04) 2004
10. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. Anal. Model. Faces Gestures **4778**, 168–182 (2007)
11. Liao, S., Law, M.W.K., Chung, A.C.S.: Dominant local binary patterns for texture classification. IEEE Trans. Image Process. **18**, 1107–1118 (2009)
12. Takala, V., Ahonen, T., Pietikäinen, M.: Block-based methods for image retrieval using local binary patterns. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) Image Analysis, vol. 3540, pp. 882–891. Springer, Berlin (2005)
13. Bougueroua, S., Boucheham, B.: Ellipse based local binary pattern for color image retrieval. In: ISKO-Maghreb: Concepts and Tools for knowledge Management (ISKO-Maghreb), 2014 4th International Symposium, pp. 1–8. Algiers, Algeria (2014)
14. Bougueroua, S., Boucheham, B.: GLIBP: gradual locality integration of binary patterns for scene images retrieval. Accepted for publication in the Journal of Information Processing Systems (JIPS), ISSN: 1976-913x(print), ISSN: 2092-805x(online), the official international journal of the Korea Information Processing Society
15. Huang, Z.-C., Chan, P.P.K., Ng, W.W.Y., Yeung, D.S.: Content-based image retrieval using color moment and Gabor texture feature. In: Ninth International Conference on Machine Learning and Cybernetics (ICMLC), 2010. Qingdao (2010)
16. Jasmine, K.P., Kumar, P.R.: Integration of HSV color histogram and LMEBP joint histogram for multimedia image retrieval. In: Intelligent Computing, Networking, And Informatics, vol. 243, pp. 753–762. Springer, India (2014)

17. Palm, C.: Color texture classification by integrative co-occurrence matrices. Pattern Recogn. **37**, 965–976 (2004)
18. Zhou, J., Xu, T., Gao, W.: Content based image retrieval using local directional pattern and color histogram. In: Optimization and Control Techniques and Applications, vol. 86, pp. 197–211. Springer, Berlin (2014)
19. Wang, X.-Y., Zhang, B.-B., Yang, H.-Y.: Content-based image retrieval by integrating color and texture features. Multimedia Tools Appl. **68**, 545–569 (2014)
20. Huang, P.W., Dai, S.K.: Image retrieval by texture similarity. Pattern Recogn. **36**, 665–679 (2003)
21. Tiecheng, S., Hongliang, L., Bing, Z., Gabbouj, M.: Texture classification using joint statistical representation in space-frequency domain with local quantized patterns. In: 2014 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 886–889 (2014)
22. Ma, W.Y., Manjunath, B.S.: Texture features and learning similarity. In: 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, pp. 425–430 (1996)
23. Swain, M.J., Ballard, D.H.: Color indexing. Int. J. Comput. Vision **7**, 11–32 (1991)
24. Rubner, Y., Tomasi, C., Guibas, L.: The Earth mover's distance as a metric for image retrieval. Int. J. Comput. Vision **40**, 99–121. 2000/11/01 2000
25. Wei-Ta, C., Wei-Chuan, L., Ming-Syan, C.: adaptive color feature extraction based on image color distributions. IEEE Trans. Image Process. **19**, 2005–2016 (2010)
26. Zhang, Q., Canosa, R.L.: A comparison of histogram distance metrics for content-based image retrieval, pp. 90270O–90270O-9 (2014)
27. Patil, S., Talbar, S.: Content based image retrieval using various distance metrics. In: Kannan, R., Andres, F. (eds.) Data Engineering and Management, vol. 6411, pp. 154–191. Springer, Berlin (2012)
28. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans. Comput. Biol. Bioinform. **1**, 24–45 (2004)