CBIR Using Combination of Color and Texture Descriptors

Caballero Daniel[1,†1,a)] Takahashi Hiroki[2,b)] Nakano Mariko[1]

Abstract: In this paper we propose a method to obtain color and texture descriptors for CBIR (Content-Based Image Retrieval) systems, this method uses a reduced HSV color space and histogram concatenation, where a color quantization method is utilized to reduce the number of colors and computational cost. The image is divided into sub blocks of equal sizes. Then the histogram of each block is computed. LBP (Local Binary Pattern) was used for texture feature. HI (Histogram Intersection) was used in both characteristics vectors to calculate the matching rates of histograms. Texture descriptor is necessary to complement the analysis of visual feature, therefore many texture methods have been done. The proposed method is evaluated on Corel1K and Outex databases with AP (Average Precision). The experimental results shown that the proposed method has a better performance than other methods that use color and texture descriptors in the same database.

Keywords: CBIR, HSV, LBP, Histogram intersection, Color quantization, Color Histogram

1. Introduction

CBIR is a technique used for extracting similar images from a database. Due to the advanced technologies the databases are increasing rapid in their size, it is difficult to index efficiently these multimedia data using the traditional metadata image and video indexing techniques. Effectively indexing and searching images in a database remains a challenge when the images are rotated, scaled, changed of illumination, etc. Considering the importance of its application, in the recent years the CBIR has become a topic of active research field [1].

The CBIR process consists in obtaining a vector that characterizes some image properties, and stored in the image feature dataset. The user provides a query image and the CBIR system computes it to get its characteristical vector and it is compared with the particular image feature database. The relevance comparison is done by using some distance measurement technique, and the minimum or permissible distances are the metrics for the matched or similar images. The features vector should be able enough to fully characterize image structural and spatial properties, which retrieve the similar images from the image database [2].

Using color histogram as a stable representation over change in view for object recognition was explored by Swain and Ballard [3], [4]. They introduced the colour indexing technique to efficiently recognize objects by matching their color histograms via HI method. Color is the most dominant and distinguishing visual feature. It is independent of image size, orientation and robust to the background, due to this in the retrieval systems the color information is considered as an essential feature. Some classical color methods include color histogram, color moment, color set and color coherent vector, etc [5]. One of the most common disadvantage on the color descriptors is the lack of spatial information.

Texture feature is an important characteristic of surfaces, scenes and objects. Texture analysis is an active area on image processing as well as pattern recognition and the fundamental for many applications such as scene recognition, biomedical image analysis, image recognition and retrieval. A great deal of work has been done on this important topic [6].

The texture-descriptors are usually computed based on the difference between a region of pixels values, they differ in their computation-method and representation [7]. There are many methods for this task, LBP is one of the most commonly used texture descriptors due to its efficiency, simple theory and computation [8]. LBP has been studied for many years, therefore multiples versions of this have surged. In this paper the basic version of LBP was used. LBP gives a good information of an image but is sensitive to the rotation and scaling, providing a different result between two images if one of them is rotated. In this paper we tried to get spatial information from a color descriptor and perform the LBP to make it robust against rotation. The combination of color and texture descriptors should be invariant to rotation and scaling.

The rest of this paper is organized as follows. In section 2 we briefly describe HI, the basic LBP and modified LBP. The proposed method is shown and we describe in detail the algorithm to obtain the descriptor in section 3. In section 4, the descriptor is tested and the experimental results are shown and compared with others descriptors. Finally the conclusions of this work are

1 National Polytechnic Institute, Mexico City, Mexico
2 The University of Electro-Communications, Tokyo, Japan
†1 Presently with The University of Electro-Communications
a) cg_daniel@outlook.com
b) rocky@hc.uec.ac.jp

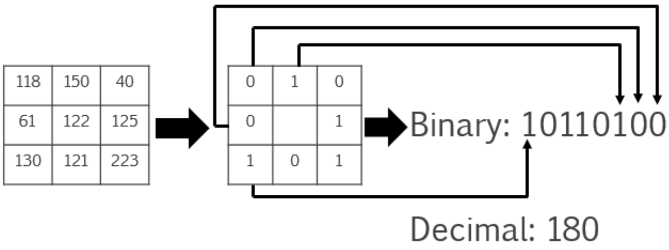


Fig. 1 Basic LBP operator

commented in section 5.

2. Preliminary

2.1 Histogram Intersection

HI was proposed by Swain and Ballar [3], [4]. Histogram Intersection matches the image color histogram with histograms of each of the models in the database. The higher the match value the better the fit to the model. The HI is given by:

$$\sum_{j=1}^{n} min(I_j, M_j). \quad (1)$$

Where j ranges over each color in the histograms, I represents the query image and M represents the model. The result of the intersection of a model histogram with an image histogram is the number of pixels from the model that have corresponding pixels of the same color in the image.

2.2 Local Binary Pattern

Local Binary Pattern operates in a local circular region by taking the difference of the central pixel with respect to its neighbors. The basic version of LBP feature of a pixel is assigned by thresholding the 3x3-neighborhood of each pixel with the center pixels value [9]. Let g_c be gray level of the center pixel graylevel and $g_p(p = 0, 1, ..., 7)$ be the graylevel of each surrounding pixel. If g_p is smaller than g_c, the binary result of the pixel is set to 0, otherwise to 1. All the results are combined to an 8-bit binary value. It is defined as

$$LBP = \sum_{p=0}^{p-1} S(g_p - g_c)2^p, S(x) = \begin{cases} 1, if g_p \geq g_c \\ 0, if g_p < g_c \end{cases} \quad (2)$$

The LBP operator takes the difference of the central pixel with the neighboring pixels and combines the values of these differences using unique weights. The order of the weights is fixed in the circular neighborhood, i.e. the weight corresponding to g_0 is always 1, for g_1 is 2, and so on, the decimal value of the binary is the LBP feature. See **Fig. 1** for an illustration of LBP.

If the image undergoes a rotation, the arrangement of the pixels around the center undergoes a shift. Since the order of the weights is fixed, the LBP computed on the rotated images is unable to deal with the rotation changes. Thus, even for a simple image rotation, the LBP operator provides very different values.

2.3 LBP modified

In order to make LBP resist to the rotation with a low computational cost, we proposed a new variant of this. it consists on take into acount the total of bins "1", the sum of the pixels will give a number $n = 0, 1, ..., 8$. See **Fig. 2** for an illustration of this method.

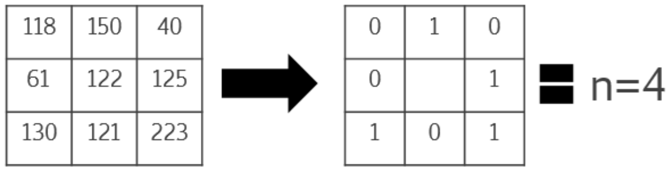


Fig. 2 Modified LBP operator

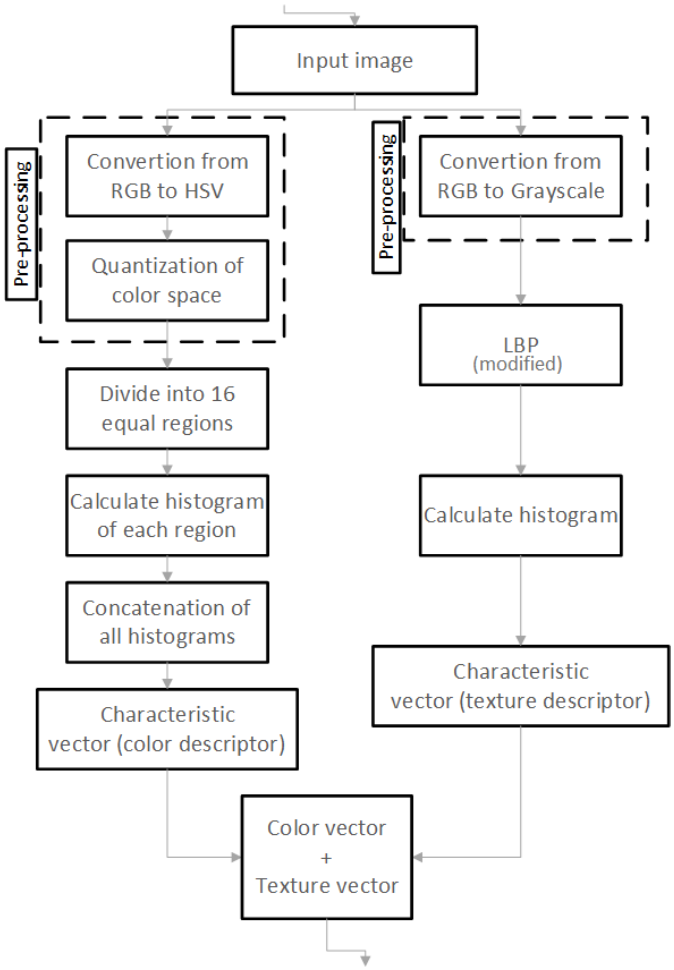


Fig. 3 Flow-Chart of proposed Descriptor

The basic LBP gives much more information than the variant of LBP, because take into account the position of the pixels and the modified LBP does not, therefore if the image is rotated the modified LBP will give the same information and this makes our propose LBP rotation invariant.

3. Proposed Method

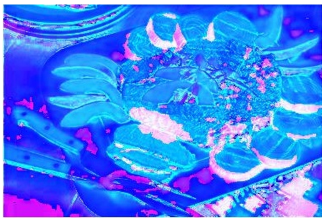
The proposed descriptor consists in, firstly in a series of steps in order to get the color and texture features as is ilustrated in **Fig. 3**. All the steps are described below.

3.1 Color Descriptor

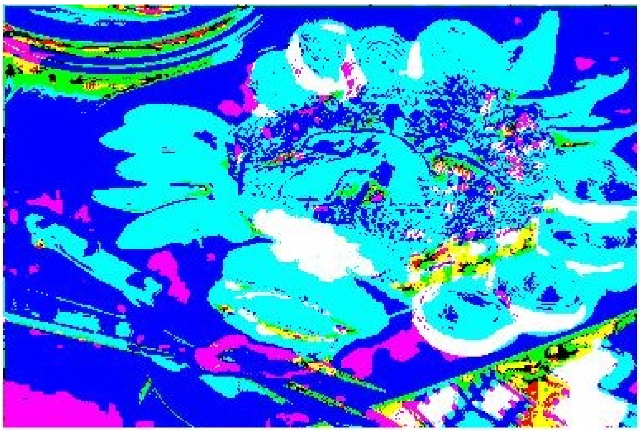
3.1.1 Convertion from RGB to HSV

HSV (Hue, Saturation and Value) is usually used because it corresponds better to how people perceive color than the RGB color space does. We transform an image RGB of size $M \times N$ to

(a) Original image



(b) HSV image



(c) HSV image 72 levels

Fig. 4 Phases on an image from RGB to HSV

HSV and quantize the HSV colors into 72 levels. An example is shown in **Fig. 4**. The transformation is as follows:

$$v = i/255, s = \begin{cases} 0 & (i = 0) \\ 1 - \frac{j}{i} & (i \neq 0) \end{cases}, h = \begin{cases} 0 & (i = j) \\ 60x\frac{g-b}{i-j} & (i = r) \\ 60x\left(\frac{b-r}{i-j} + 2\right) & (i = g) \\ 60x\left(\frac{r-g}{i-j} + 4\right) & (i = b) \end{cases} \quad (3)$$

Where i represents the maximum value of (r, g, b) and j the minimum value of (r, g, b).

Then the color image is uniformly quantized into 72 levels. Specifically the H, S and V color channels are uniformly quantized into 8, 3 and 3 bins, respectively. The next step is construct the color feature of the image into one-dimension, for this we use the next equation:

$$P = Q^2H + QS + V \quad (4)$$

Where $Q = 3$ and represent the number of the quantization of color space chanel S and V [10].

3.1.2 Dividing the image into $n \times m$ sub-blocks and obtaining Histogram

The image is divided into $n \times m$ equal regions as shown in **Fig. 5**, the algorithm works with any size of image. Then we calculate the histogram of each sub-block and all the histograms are concatenaded as shown in **Fig. 6** and **Fig. 7**. This provides a more distinctive set of values for an image. Therefore, these concatenation from the local regions give a robust information from the histograms.

Finally our characteristical vector also know as descriptor of color, is obtained. The retrieval matching is done by using the HI in the histogram of query image with the histogram of target image in corresponding position. The results are normalized between 0 to 1, the images closest to 0 are the more similar to the query.

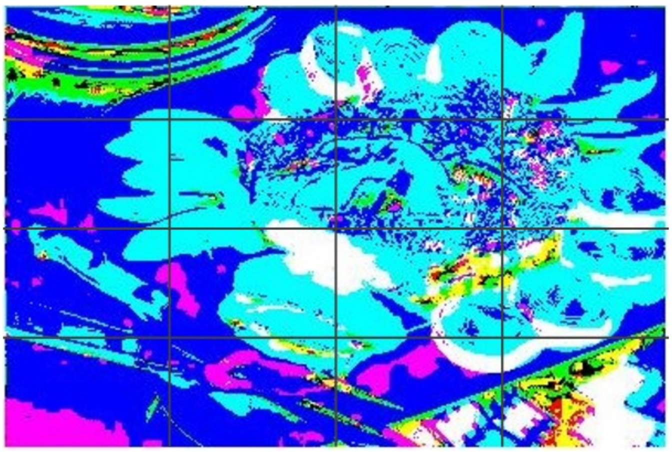


Fig. 5 Image divided into 4x4 sub-blocks

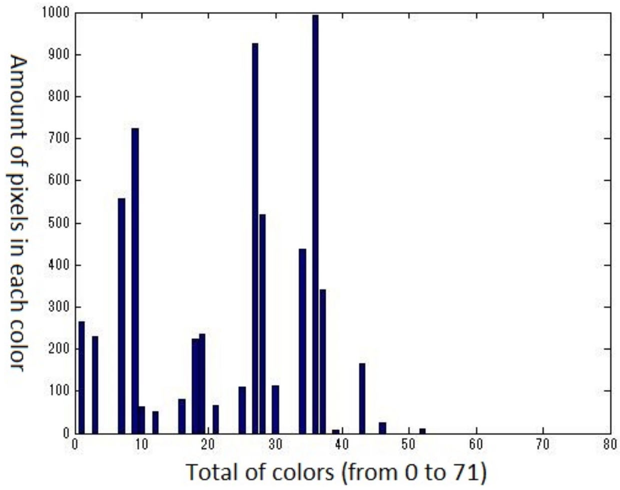


Fig. 6 Histogram of a region

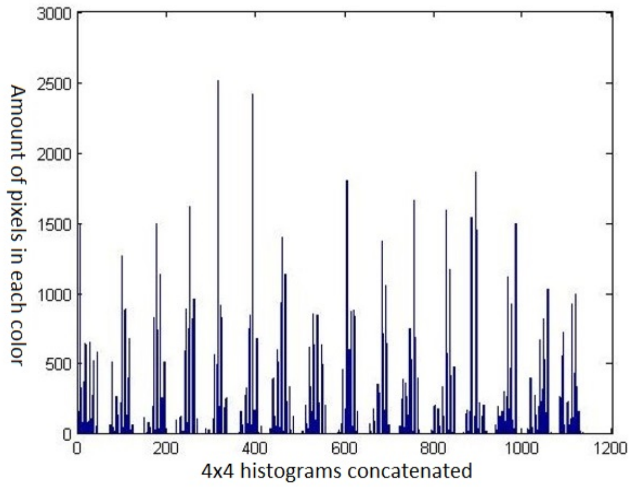


Fig. 7 Histogram of all regions

3.2 Texture Descriptor

In order to get this descriptor the LBP was aplied in an grayscale image, we used both LBP and modified LBP methods described in section 2 to compare the performance. In the case of the basic LBP a vector with a length of 256 elements is obtained and for the proposed LBP we get a vector with a legth of 9 elements.

3.3 Combination of Vectors

In CBIR systems, given a query image, the most relevant images are extracted from a database using some of many distance metrics between an image query and image from a database. The literature says that the relevant images have smaller distance from the query image. Due to the result of the different distance metrics, the rank of them are variable. Therefore a normalization of the values is necessary to combine two or more descriptors. From the weighted linear combination method, two visual features, color and texture are combined lineally, the global distance is given by

$$D = w_c d_c + w_t d_t \tag{5}$$

Where d_c and d_t are distances of color and texture features, respectively, and w_c and w_t are corresponding weight values which must satisfy $w_c + w_t = 1$ [11].

4. Experimental Results

To evaluate the performance of the proposed color and texture descriptors, we use the Corel1K dataset [12], [13] that is composed by 1000 color images, these images are classified into 10 categories, which are Tribe, Beach, Buildings, Buses, Dinosaurs, Elephants, Roses, Horses, Mountains and Food.

The Outex TC_00010 dataset was also used, it is composed by 4320 grayscale images, these images are classified into 24 categories with 180 images per category, with spatial rotations of $00, 05, 10, 15, 30, 45, 60, 75, 90$ degrees. The images are of size of 128×128 with a resolution of 100 dpi [14]. This dataset is special to extract texture feature and was used to evaluate the LBP descriptor.

4.1 Performance Measurement

The performance measurement is based on the quantity of images retrieved successfully given a query image. In this context the most commonly used measurement is the average precision (AP) and average normalized retrieval rank (ANMRR). AP is given by

$$AP = \frac{\text{Number of relevant images retrieved}}{\text{Total of images retrieved}} \tag{6}$$

ANMRR is considered one of the more accurate in the CBIR systems, it combines many metrics such as success and no success counters, as well as the rank of the images retrieved. ANMRR is given by

$$ANMRR = \frac{1}{NQ} \sum_{q=1}^{NQ} NMRR(q) \tag{7}$$

$$NMRR = \frac{2AVR(q) - N_G(q) - 1}{2W - N_G(q) + 1} \tag{8}$$

$$AVR(q) = \frac{1}{N_G(q)} \sum_{k=1}^{N_G(q)} R(k) \tag{9}$$

Where NQ is the total number of classes, $N_G(q)$ is the number of ground truth images per class, W is maximum windows size, which is $2N_G(q)$. $R(k)$ belongs to the rank of ground truth for a query q in the first W images. For the images that are not retrieved within W images, $R(k) = W + 1$.

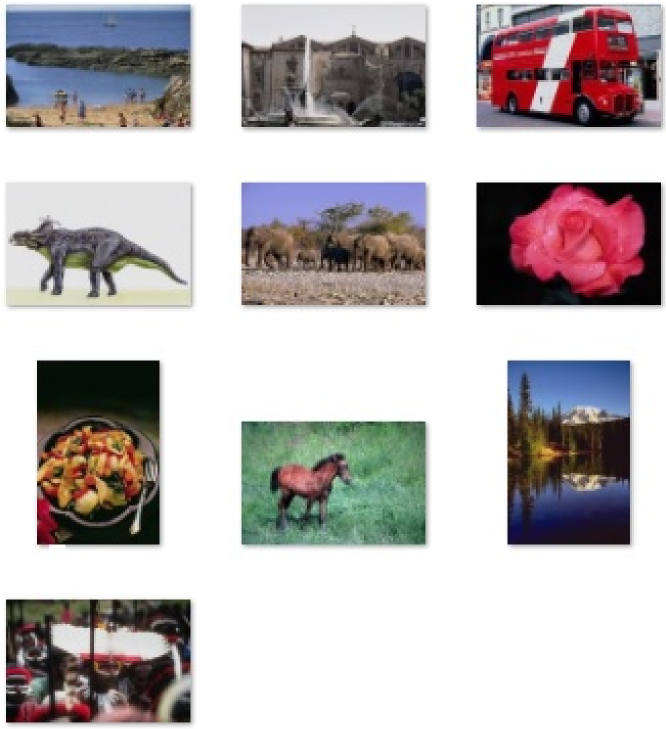


Fig. 8 One sample image from each of 10 classes in Corel1K dataset.

The range of AP is $[0, 1]$, the large value indicates good performance, while smaller value indicates the deficiency of the descriptor used for this task. Conversely in ANMRR the smaller value indicates good performance, while large value means the deficiency of the descriptor used in this task.

4.2 Performance Comparison Color Descriptor

To evaluate the proposed descriptor we used 80 out of 1000 images, 8 images per category, as query images, which are same images used in [15]. The literature establishes that for having consistent results the number of queries must be at least 1% of the dataset size [16] . In our experiments, each image is taken as a query, the number of queries images represents the 8% of the dataset, see **Fig. 8**.

In order to know the number of regions to be divided the image, we compare diferents regions with AP in Corel1K database and the 4×4 regions gives the best final result againts other number of region, we can see these comparison results in **Table 1**.

In the **Table 2** it can be seen the comparison of average precision obtained by the proposed method with other retrieval systems. As we can see our proposed method improves the relevant image extraction generally. some of the methods which we are comparing used two low level features. In the **Table 3** we can see other way to show the performance by apliying ANMRR, as we said previously as close to zero is better performance.

4.3 Performance Comparison Texture Descriptor

To evaluate the LBP descriptor we used all 4320 images to test the performance. In our experiments each image in the database is

Table 1 Differents sizes of divided regions

Class Name	2×2	3×2	3×3	4×3	4×4	5×4	5×5	6×5	6×6	7×6	7×7	8×7	8×8
Tribus	0.6862	0.6987	0.7094	0.6480	0.6588	0.6473	0.6192	0.6164	0.6121	0.6016	0.5891	0.5597	0.5434
Beach	0.5980	0.5404	0.5579	0.5394	0.5336	0.5147	0.5123	0.5007	0.5028	0.5255	0.5274	0.5213	0.5242
Building	0.3877	0.3943	0.4118	0.4261	0.4200	0.4198	0.4295	0.4215	0.4260	0.4267	0.4259	0.4347	0.4464
Bus	0.7242	0.7304	0.7404	0.7315	0.7488	0.7449	0.7350	0.7368	0.7329	0.7154	0.7230	0.7263	0.7097
Dino	0.9297	0.9273	0.9392	0.9359	0.9409	0.9421	0.9438	0.9454	0.9418	0.9414	0.9427	0.9463	0.9488
Elephant	0.4906	0.4899	0.4965	0.5252	0.5207	0.5179	0.5226	0.5041	0.5037	0.5158	0.5188	0.5075	0.5078
Flower	0.6739	0.7141	0.7637	0.7704	0.7886	0.7884	0.8039	0.7971	0.7969	0.8084	0.8099	0.8154	0.8160
Horse	0.6227	0.6629	0.6547	0.6839	0.6840	0.6763	0.6768	0.6721	0.6775	0.6887	0.6853	0.6805	0.6965
Mountain	0.3627	0.3722	0.3860	0.3973	0.4001	0.3841	0.3880	0.3922	0.3959	0.3883	0.3902	0.3874	0.3889
Food	0.8541	0.8419	0.8343	0.8420	0.8306	0.8374	0.8192	0.8211	0.8297	0.8202	0.8182	0.8038	0.7933
Average	0.6330	0.6372	0.6494	0.6500	0.6526	0.6473	0.6450	0.6407	0.6419	0.6432	0.6431	0.6383	0.6375

Table 2 Comparison of Average Precision

Class Name	H. A. Jalab, 2011 [15]	J. Pujari and P. Hiremath [17]	Ch.Kavitha, B.P. Rao and A.Govardhan, 2011 [2]	A. Hafiane and B. Zavidovique, 2008 [18]	A. N. Fierro, M. Nakano and H. Perez, 2013 [19]	Proposed color method.
Tribe	32.3%	54%	41%	44.1%	61%	**65.9%**
Beach	**61.2%**	38%	32%	30.6%	46.6%	53.4%
Buildings	39.2%	40%	37%	38.2%	41.1%	**42%**
Bus	39.5%	64%	66%	67.6%	62.9%	**74.9%**
Dinosaur	**99.6%**	96%	43%	97.2%	93.5%	94.1%
Elephant	55.7%	**62%**	39%	33.8%	41.8%	52.7%
Flower	**89.3%**	68%	87%	88.8%	77.5%	78.9%
Horse	65.2%	75%	35%	63.2%	**77.4%**	68.4%
Mountain	**56.8%**	45%	34%	31.3%	42.8%	40%
Food	44.1%	53%	31%	34.9%	68.6%	**83.1%**
Average	58.29%	59.5%	44.5%	52.97%	61.3%	**65.3%**

Table 3 Results of ANMRR

Class Name	NMRR
Tribe	0.51
Beach	0.55
Buildings	0.59
Bus	0.36
Dinosaur	0.10
Elephant	0.58
Flower	0.24
Horse	0.35
Mountain	0.68
Food	0.27
ANMRR	0.42

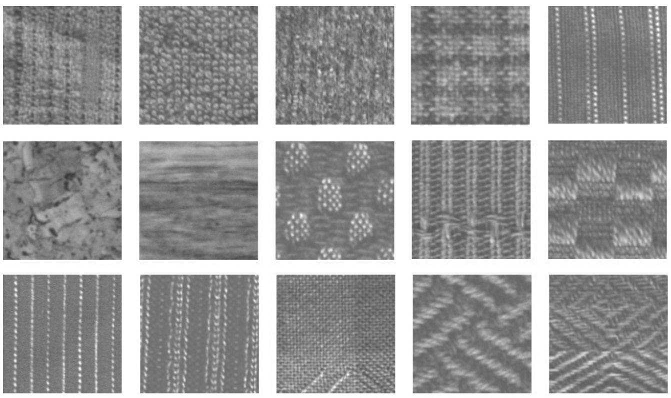


Fig. 9 One sample image from classes in Outex dataset.

taken as a query and matched to all the images in the database (including itself) the number of queries images represents the 100% of the dataset, see **Fig. 9**.

The result of LBP descriptor is a comparison between the basic LBP against our proposed LBP, these results can seen in **Table 4**. As we can see the proposed method appears be invariant to the rotation being better against the basic LBP.

We also tested the two LBP descriptors in the Corel1k database, in this case we can see how the basic LBP has better

Table 4 AP comparison of two LBPs on Outex database

Classes	Modified LBP	Basic LBP
1	0.957	0.811
2	0.907	0.803
3	0.910	0.934
4	0.828	0.630
5	0.944	0.753
6	1.000	0.926
7	0.886	0.770
8	0.996	0.877
9	0.957	0.623
10	0.753	0.482
11	0.832	0.611
12	0.718	0.650
13	0.653	0.713
14	0.642	0.485
15	0.326	0.398
16	0.578	0.542
17	0.647	0.490
18	0.607	0.424
19	0.635	0.710
20	0.640	0.872
21	0.932	0.864
22	0.898	0.662
23	0.928	0.872
24	0.912	0.808
Average	0.795	0.696

performance, it may be due to the database has not rotated images and this method contain more information from the images. In **Table 5** we can also see the results of the color descriptor.

4.4 Performance Comparison of the combination of Color and Texture Descriptors

For a good comparison with the literature [10] we use the same conditions, each image in the database is taken as a query and matched to all the images in the database (including itself) and we retrieved 20% of the ground truth, therefore the number of queries is 100%. For the combination we use both textures descriptors basic LBP and modified LBP. Basic LBP apears be the

Table 5 AP Comparison of LBP methods on Corel1k

Class Name	Proposed LBP	Basic LBP	Color descriptor
Tribus	0.5583	0.6539	0.6470
Beach	0.3917	f0.5996	0.5407
Building	0.4458	0.5848	0.6280
Bus	0.8545	0.9295	0.7891
Dino	0.9531	0.9811	0.9990
Elephant	0.4154	0.5046	0.7255
Flower	0.8020	0.8866	0.8542
Horse	0.7043	0.7752	0.9258
Mountain	0.4347	0.4280	0.4977
Food	0.5038	0.6188	0.7091
Average	0.6064	0.6962	0.7316

Table 6 Comparison between two color and texture based descriptors

	B. Nan, Y. Xu, Z. Mu and L. Chen, 2015 [10]	Color descriptor	Color + modified LBP	Color + basic LBP
Weight			$w_c = 0.6$, $w_t = 0.4$	$w_c = 0.6$, $w_t = 0.4$
Tribus	0.774	0.6470	0.7452	0.7935
Beach	0.3670	0.5407	0.5794	0.6412
Building	0.6315	0.6280	0.6715	0.6959
Bus	0.7855	0.7891	0.8872	0.9239
Dino	0.9880	0.9990	0.9964	0.9977
Elephant	0.6395	0.7255	0.7445	0.7748
Flower	0.8775	0.8542	0.9296	0.9392
Horse	0.9070	0.9258	0.9187	0.9297
Mountain	0.3400	0.4977	0.5471	0.5913
Food	0.7517	0.7091	0.7592	0.8006
Average	0.7061	0.7316	0.7779	0.8088

best due to the retrieval results, the method uses color and texture features for the retrieval. The results are shown in **Table 6**.

5. Concluding Remarks

In this paper we proposed a new color-based descriptor with the combination of LBP descriptor for the CBIR task in order to improve the retrieval performance. This method is based on HSV and HI. HI by itself does not take into account local spatial properties. Partitioning the image in sub-blocks, our descriptor takes into account the global and local distribution of colors in a given image. Using Corel1K and Outex dataset we demonstrated the efficiency of our method against others method that used color and texture features. The results show that if we aplied a robuster texture feature descriptor the combinations will become stronger. As further studies, the proposed retrieval method will be combined with shape feature and evaluated.

Acknowledgments The authors thanks the National Polythecnic Institute of Mexico and the University of ElectroCommunications for the financial support in ths work.

References

[1] Hu, W., Xie, N., Li, L., Zeng, X. and Maybank, A.: A survey on visual content-based video indexing and retrieval, *IEEE Trans. on Systems, Man, and Cybernetics-Part C, vol. 41, no. 6, pp.797-819 (2011).*

[2] Kavitha, Ch., Prabhakara, B. and Govardhan, A.: Image retrieval based on color and texture features of the image sub-blocks, International Journal on Computer Applications, vol. 15, no. 7, pp.33-37 (2011).

[3] Swain, M.J. and Ballard, D.H.: Indexing via colour histograms, Proceedings of the Third International Conference on Computer Vision, pp.390-393 (1990).

[4] Swain, M.J. and Ballard, D.H.: Colour indexing, International Journal of Computer Vision, pp.11-32 (1991).

[5] Pass, G., Zabih, R. and Miller, J.: Comparing images using color coherence vector, Proceedings of the fourth ACM international conference on Multimedia, Boston, Massachusetts. City of Publisher, USA. ACM, doi: 10.1145/244130.244148 (1997).

[6] He, J., Ji, H. and Yang, X.: Rotation invariant texture descriptor using local shearlet-based energy histograms, IEEE Signal Processing Letters, pp. 905-908 (2013).

[7] Sharma, M. and Ghosh, H.: Histogram of gradient magnitudes: A rotation invariant texture-descriptor, 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, pp. 4614-4618 (2015).

[8] Xiao, J. and Wu, G.: A Robust and Compact Descriptor Based on Center-Symmetric LBP, 2011 Sixth International Conference on Image and Graphics (ICIG), Hefei, Anhui, pp. 388-393 (2011).

[9] Ojala, T., Pietikäinen, M. and Harwood, D.: A comparativestudy of texture measures with classification based on featured distributions, Pattern recognition, Vol.29, No. 1, pp. 51-59 (1996).

[10] Nan, B., Xu, Y., Mu, Z. and Chen, L.: Content-based image retrieval using local texture-based color histogram, International Conference on Cybernetics(CYBCONF), Gdynia, pp.399-405 (2015).

[11] Fierro, A. N., Toscano, K., Nakano, M., Perez, H., Cedillo, M. and Garcia, F.: Automatic visual features weights obtention for content-based image retrieval systems , Computing Science and Automatic Control (CCE), 2015 12th International Conference on Electrical Engineering, Mexico City, pp.1-5 (2015).

[12] Wang, J. Z., Li, J. and Wiederhold, G.: SIMPLIcity: Semanctics-sensitive integrated matching for picture libraries, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 9, pp.947-963 (2001).

[13] Li, J. and Wang, J. Z.: Automatic linguistic indexing of pictures by a statistical modeling approach, IEEE Transactions on Pattern Analysis and Machine Intelligence, *Vol. 25, No. 9, pp.1075-1088 (2003).*

[14] University of Oulu: Outex texture classification database, available from ⟨http://www.outex.oulu.fi/index.php?pa- ge=classification⟩ (accessed 2014-11-21).

[15] Jalab, H. A.: Image retrieval system based on color layout descriptor and Gabor filters, *IEEE Conference on Open System, Langkawi, Malaysia, pp.32-36 (2011.)*

[16] Talamantes, J. S., Cruz, C. A., Cortez, J. V. and Azuela, J. S.: Self organizing natural scene image retrieval, Expert Systems with Applications, vol. 40, pp.2398-2409 (2013).

[17] Pujari, J. and Hiremath, P.: Content-based image retrieval based on color, texture and shape features using image and its complement, International Journal of Computer Science and Security, vol. 1, issue 4.

[18] Hafiane, A. and Zavidovique, B.: Local relational string and mutual matching for image retrieval, Information Processing & Management, Vol. 44, pp.1201-1213 (2008).

[19] Fierro, A. N., Perez, H. and Nakano, M.: An efficient color descriptor based on global and local color features for image retrieval, International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE) , Mexico City, Mexico. pp. 233-238 (2013).