A Compact Color Descriptor for Image Retrieval

Vedran Ljubovic, Haris Supic
University of Sarajevo, Faculty of Electrical Engineering Sarajevo
Zmaja od Bosne bb
71000 Sarajevo, Bosnia and Herzegovina
vljubovic@etf.unsa.ba, hsupic@etf.unsa.ba

***Abstract*—The resource usage in Content-Based Image Retrieval is a frequently neglected issue. This paper describes a novel compact feature vector based on image color histograms in the HSL color space. The images are represented using only 10 bytes per image. It is shown that, in the context of Query-by-Example (QbE) usage scenarios, the method described achieves retrieval performance close to the state of the art image retrieval methods that use considerably more memory. It is also shown that the described method outperforms other methods with similar memory usage.**

Keywords*—content-based image retrieval, histograms, histogram distance, color spaces, HSL, Matsushita distance, query-by-example

I. Introduction

A crucial element of any Content-Based Image Retrieval (CBIR) system is the feature extraction, a process that attempts to represent a computer image with a limited amount of data. The created feature vector database represents collections of images. In the Query-by-Example (QbE) application such database can be searched for images that are most similar to the given query image.

Most CBIR systems include a feature vector that represents image coloration. Color histograms are one of the earliest and best known such color features [17].

A large number of papers focus on getting the optimal retrieval performance using metrics common in information retrieval science such as precision and recall. There are relatively few papers that focus specifically on the feature vector size, such as [3]. Resource usage in CBIR systems is still an important issue, even with the progress of computing technology [6].

This paper explores a novel color descriptor that can be used alone or in combination with other descriptors for texture, layout etc. It features fast calculation and has low memory usage. The paper also gives optimal approach to calculating feature distance. The latest datasets and performance metrics are used to compare our approach to other methods mentioned in literature, both state-of-the-art ones and those specifically designed for the lower memory usage. Overall the described retrieval method seems deceptively simple, but a large number of experiments (over 4000) were performed to tune all the parameters to reach the achieved result.

The remainder of this paper is structured as follows: Chapter II gives an overview of the previous work in this area. Chapter III gives a detailed description of the algorithms and methods used. Chapter IV explains the experimental methodology and apparatus, and presents experimental results in a brief discussion. Finally, Chapter V gives the conclusion and some directions for the future work.

II. Previous Work

It is long known that large feature vectors created by the common retrieval methods are not usable in practical image retrieval scenarios. In 2001 the MPEG-7 working group has recommended a number of image retrieval compact descriptors that can be used individually or in combination [9]. The MPEG-7 effort also contributed to the image retrieval method comparison methodology.

After this seminal work, researchers focused on benchmarking and suggesting improvements to MPEG-7, for example in [20]. The paper [5] proposes another novel method based on fuzzy processing of color histograms, while [2] combines such histogram with texture for a compact image descriptor that combines both types of features. In [3] the same authors propose an improved compact composite descriptor that provides the best overall performance of all tested descriptors.

The earliest work on using color histograms in image retrieval is [17] which uses histograms in RGB color space. The RGB model is convenient for image representation in computers because it models the way images are displayed on CRT monitors. A number of other papers on color histograms also use RGB color model (e.g. [4]).

However, RGB model doesn't correspond to the way humans perceive color. HSV color space is explicitly designed to model human color perception, and is therefore used in most papers on histograms as a global feature in CBIR (e.g. [3], [16], [21], [12]). It is also used for the Scalable Color Descriptor (SCD) feature of the MPEG-7 standard for multimedia content description [9].

A closely related color model is HSL, however there are few papers that use this particular color model in the image retrieval. It is commonly – and mistakenly – assumed that this model is identical to the HSV for the purposes of image retrieval.

None of the above color spaces is uniform. Papers [5] and [14] suggest that using a uniform color space such as CIE L*a*b* or L*u*v* should deliver superior retrieval performance. However, conversion from RGB to Lab requires an extra step of conversion to XYZ model and a calculation of cube root, making it computationally expensive [3].

Another problem with the color histograms in image retrieval is that of dimensionality reduction. Computer images are usually represented using 24 bits per pixel. Resulting histogram with 16,7 million bins would typically use more memory than the image itself. Most papers use some approach to reduce the number of bins.

The most common approach since the earliest work on histograms [17] is the following: Each of color channels is first quantized by taking a number of most-significant bits. Then these bits are concatenated to form an index into a single histogram [3], [16]. Such histogram is called linked.

For example, a commonly used quantization scheme is HSV 16*4*4 as is the case with MPEG-7 Scalable Color Descriptor [9]. Here, value for Hue (H) is quantized as a 4-bit integer (value in range [0,15]), while Saturation (S) and Value (V) are quantized as 2-bit integers (values in range [0,3]). The resulting values are combined to form an index I to a single big histogram using the following formula:

$$I = 16 \cdot H + 4 \cdot S + V$$

We see that I is an 8-bit integer (value in range [0,255]) and therefore a histogram with 256 bins is obtained.

Sometimes heuristic can be used to further reduce the number of colors. For example, in the HSV color model, gray can be described as any pixel where saturation (S) is below certain threshold [16]. Similarly, black is a color where value (V) is below certain threshold. However, white color characterization requires observation of both S and V. In HSL color model white is described simpler: any pixel where lightness (L) is above the given threshold. Another problem with the HSV model is that the human perception of color (hue) varies with saturation [16]. This problem likewise is less pronounced in HSL model.

The hue can be quantized by a formula [12], however a better approach is to use the table of values for hue and their mapping to perceived colors [21]. Results can be further improved by using fuzzy color processing [3], [5].

The issue of how histograms are represented in memory i.e. how many bits are used to represent each bin is surprisingly frequently omitted from papers. It seems that most papers assume using the native floating point type of a programming language, typically of 32 bits size. MPEG-7 SCD uses only one bit per bin, which is processed using Haar transform [9]. Our analysis of source code to LIRe, the open-source image retrieval platform [8] revealed that its color histogram feature is linearly quantized to 10 bits per bin.

A conclusion reached after a large number of experiments is that, when using linear quantization of histogram bins, decreasing the number of bits per bin results in a rapidly decreased retrieval performance [7]. Optimal number of bits per bin is between 8 and 12 bits. However some papers use nonlinear quantization. In CEDD and FCTH histogram is quantized to 3 bits per bin using coefficients obtained using Gustaffson Kessel fuzzy classifier [3].

Finally, there must be a method for calculating distance between two histograms. Most relevant papers use the Manhattan distance (L1) or the Euclidean distance (L2), although later work suggests that the optimal distance metrics for histograms may be the Jeffrey divergence (Jensen-Shannon divergence) [4] or histogram intersection [12], [14]. A comprehensive study on the distance metrics for color histograms [7] found that using the Matsushita distance gives the best results, especially when using a large number of bits per bin and a small number of bins.

Table I. EXPERIMENTALLY OBTAINED OPTIMAL QUANTIZATION FOR HUE AND ITS INTERPRETATION IN TERMS OF VISIBLE COLORS

Hue (0-360)	English language color name
0-20	red
20-70	yellow
70-140	green
140-210	cyan
210-240	blue
240-280	magenta
280-360	pink

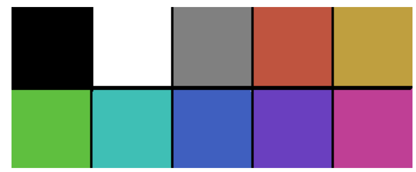

Figure 1. The palette of 10 basic colors used in the method described in the paper

III. DESCRIPTION OF RETRIEVAL METHOD

Feature extraction starts with the image conversion to HSL (Hue-Saturation-Lightness) color model. Computer images are typically represented using the RGB color model. Conversion to HSL is fast and direct compared to perceptually uniform color models such as the CIE L*a*b* used in [5]. Added complexity of conversion to the CIE L*a*b* is not justified by improved retrieval performance [3].

An important feature of the HSL color model is that black, white and gray can be characterized by extreme values of individual components (Lightness or Saturation respectively). In the more common HSV model, characterization of white color requires that both Saturation be low and Value be high. Another consequence is that in the HSV model the effect of saturation on perceptible color can't be ignored as is done in this paper.

Upon conversion, the color of each pixel is quantized as follows: If Lightness is lower than a black threshold (BT) the color is black. If it is higher than a white threshold (WT) the color is white. Otherwise, if Saturation is lower then a gray threshold (GT) the color is gray. If none of the above rules apply, Saturation and Lightness are discarded and Hue is quantized into one of the seven values representing seven basic colors.

Through a large number of experiments that can be found on authors' website[1], the best retrieval results were obtained with BT=15, WT=95 and GT=15 assuming that S and L are values on interval [0,100]. The optimal quantization for Hue and its interpretation is given in Table I.

Thus the whole image is represented using 10 basic colors given in Fig. 1.

[1] http://people.etf.unsa.ba/~vljubovic/etfis/

However, there are still certain nonlinearities in HSL color perception. For example, let's observe a pixel with H=100 (green), S=20 and L=95. A human observer would not notice any coloration in this pixel, i.e. it would be recognized as somewhere between gray and white, closer to white. However, if we increase S to 100, the pixel gets an unmistakable green tint. Similar tint is obtained if we keep S at 20 but reduce L to 80.

To address these nonlinearities, fuzzy color matching inspired by [3], [5], [2] is used. A set of 20 rules is used with fuzzy antecedents and crisp consequents evaluated using the Multi-Participant algorithm. First the Lightness (L) is evaluated to determine membership to the Black and White bins. Then the Saturation (S) is evaluated for membership to the Gray bin. Finally, the Hue is evaluated for membership into 7 colors.

Through a large number of experiments, ideal values for fuzzy limits were established. The BT thus became a fuzzy value [6,17], WT is [60,96], while GT is [10,21]. Each of the color transitions in the Table I is given a 20 degree fuzzy zone. For example, colors with the Hue in range [0,10] are classified as red, [10,30] are between red and yellow, [30,60] is yellow, etc.

A histogram is formed of these 10 basic colors having 10 bins. Upon calculation the histogram is normalized and quantized to 8 bits per bin using the following formula:

$$h_N(i) = \lfloor \frac{h(i)}{n} \cdot 2^8 \rfloor$$

where h is the starting histogram, n is the total number of pixels in image, and h_N is the normalized and quantized histogram.

The following important issue in information retrieval is that of calculating distance between feature vectors. The retrieval method presented in this paper uses the Matsushita distance which is defined with the following formula:

$$D = \sqrt{\sum_i (\sqrt{x_i} - \sqrt{y_i})^2}$$

Through a large number of experiments available on authors' website it was found that the Matsushita distance delivers superior performance to other distance metrics such as L1 distance, L2 distance, Jensen-Shannon divergence [4], histogram intersection [12], [14], etc.

IV. Experimental Methodology and Results

Use of standardized, annotated datasets is an important issue in the CBIR [10]. To this purpose, we used three datasets consisting of photograph collections that are often cited in literature: Wang SIMPLYcity [19], UW [1] and UCID [15].

Wang dataset consists of 1000 images in 10 categories (100 images each). The results are considered relevant if they are in the same category.

UW dataset is accompanied with a "ground truth" file where all or most of the images are annotated using several words (tags) describing image content such as "trees", "sky", "buildings" etc. A result is considered relevant if any of the tags are common between the query image and the result image.

Finally, the UCID dataset includes a "ground truth" file where a set of relevant results is given for each query image. For many images in this dataset only one relevant result is given. The consequence is that most CBIR methods exhibit low precision but high recall in this dataset.

Measuring is done through an exhaustive search i.e. each image in the dataset is used as a query image (images with no ground truth are skipped). The query image itself is omitted from the results.

A common retrieval methods comparison metric is the Precision-Recall graph. Two methods are compared by overlaying their graphs [11]. This, however, is impractical for comparing a large number of retrieval methods in many configurations across several datasets. Preferred approach is to have a simple numerical metric for each query, which is then averaged across a large number of queries. After evaluating literature, we chose the following metrics:

Mean Average Precision (MAP) - mean value for precision is calculated across the 11 usual points in Precision-Recall graph [11], [18]. This is a standard measure used by NIST in its TREC contest.

AP15 - average value for precision in the first 15 retrieved images [11]. This metric simulates a search engine which returns results in pages of 10-20 results. Research suggests that most users don't look beyond the first page [13].

Average Normalized Modified Retrieval Rank (ANMRR) is explained in [9]. This is the standard metric used in the ImageCLEF competition.

In addition we wanted to gauge the ability of tested methods to quickly retrieve results of higher relevance, especially in UW dataset. To this purpose we introduce relevance score as number of tags that are common between query image and retrieved image, divided by number of tags in query image:

$$Relevance = \frac{Nr.common\ tags}{Nr.query\ tags}$$

Relevance is a value in range [0,1] where the higher value corresponds to the higher relevance of each result. Thus weighted precision (WP) can be calculated using the following formula:

$$WP = \frac{\sum_{i=1}^{N} Relevance_i}{N}$$

where N is number of retrieved images. This allows us to calculate average weighted precision (AWP15) as a metric comparable to AP15 given above.

The method presented in this paper is compared to a number of CBIR methods described in literature.

The most common type of the color feature is histogram in the RGB color space where the three color components are combined (or linked) so that the three most significant bits are used from each component and then concatenated giving a total of 9 bits. Thus the resulting histogram has 512 bins.

Table II. EXPERIMENTAL RESULTS. FOR MAP AND AWP15 METRICS MORE IS BETTER, FOR ANMRR LESS IS BETTER. RIGHT COLUMN FOR EACH METRIC GIVES RANKING (E.G. 1 IS BEST PERFORMING METHOD). TIMES ARE IN SECONDS. “EXPER. TIME” IS TIME REQUIRED TO PERFORM THE EXPERIMENT (EXHAUSTIVE SEARCH IN ALL DATASETS).

Dataset	WANG				UW						UCID				Ft. size	Index time	Exper. time
Method	MAP		ANMRR		MAP		AWP15		ANMRR		MAP		ANMRR				
This paper	0.4977	2	0.4188	4	0.6307	1	0.3738	6	0.2601	3	0.3621	1	0.5612	1	10	79	83
RGB	0.4733	5	0.4274	5	0.6114	6	0.3820	3	0.2643	7	0.2369	5	0.6380	5	512	36	54
MPEG-7 SCD	0.3108	8	0.5994	8	0.6048	7	0.3210	8	0.2604	4	0.1223	8	0.8272	8	~32	59	58
MPEG-7 CLD	0.4365	7	0.4684	7	0.5971	8	0.3239	7	0.2819	8	0.1683	6	0.7557	7	8	36	40
CEDD	0.4982	1	0.4080	1	0.6155	5	0.3851	2	0.2615	6	0.2718	3	0.5915	3	54	94	100
C.CEDD	0.4872	4	0.4183	3	0.6186	3	0.3797	4	0.2600	2	0.2547	4	0.6133	4	23	93	93
FCTH	0.4906	3	0.4107	2	0.6185	4	0.3936	1	0.2604	4	0.2742	2	0.5892	2	72	112	106
C.FCTH	0.4699	6	0.4277	6	0.6198	2	0.3768	5	0.2577	1	0.1287	7	0.7472	6	30	101	109

The histogram is then quantized to 10 bits per bin. The L1 distance is used for distance metric.

A very compact color feature based on the color histograms is Scalable Color Descriptor (SCD) which is a part of MPEG-7 specification for multimedia content description [9]. Another compact feature used in the MPEG-7 is Color Layout Descriptor (CLD) which is not derived from the color histograms but uses Discrete Cosine Transform (DCT) to create a scheme of spatial distribution of color in an image. SCD and CLD are evaluated using implementations given in the LIRe open-source image retrieval platform [8]. The configuration for SCD is the default used in LIRe: 256 coefficients, resulting in cca. 32 bytes per image whereas CLD always uses 8 bytes per image [20].

There are several proposals for improving MPEG-7 performance maintaining a compact vector. We used CEDD [3] and FCTH [2], as well as their compact versions – C.CEDD and C.FCTH, as state-of-the-art methods for comparison with our method. These methods combine color histograms using 24 bins (standard versions) and 10 bins (compact versions) with texture features (FCTH and C.FCTH) and edge histogram (CEDD and C.CEDD) respectively. Again, implementations in LIRe were used in experiments.

The experimental results are presented in Table II. For the MAP and AWP15 higher values mean better results, while for the ANMRR lower values correspond to better retrieval. In addition to performance metrics described above, feature size and timing is given. The total time for indexing all three datasets and for performing all experiments (exhaustive search) is given on an Intel Core i5 PC. A highly efficient implementation in C++ is used for the method presented in our paper.

The Table II shows that the method presented in this paper outperforms the common RGB histogram which uses exponentially larger amount of memory. Our method also consistently outperforms the similar MPEG-7 SCD, as well as the MPEG-7 CLD. The state-of-the-art CEDD and FCTH methods deliver better retrieval performance at greater memory usage, however it must be noted that these methods combine color information with texture, whereas method in this paper uses just color information.

V. CONCLUSIONS AND FUTURE WORK

The method presented in this paper does not give the best possible retrieval performance, nor is it unique in its small feature vector size. However, it does represent a very reasonable compromise between the two requirements. The best overall results are obtained using CEDD, followed by FCTH. However, these methods have considerably larger feature size and indexing time compared to method presented in this paper.

The principal contributions of this paper are: use of the HSL color space and corresponding new coefficients for fuzzy color matching, and use of the Matsushita distance. Using latest methodology and datasets, we have shown that the method presented in this paper gives superior performance to other methods that have similarly small feature vector. It was also shown that this method gives results comparable to retrieval methods that use considerably more memory.

Therefore, it should be interesting to combine the color feature described here with some texture feature in the manner similar to that described in [3].

Given very low resource usage for this method it should be especially interesting to apply such approach in JPEG compressed domain [6].

Another problem that should be addressed is that of the efficient search of feature vectors. It is evident from timings in Table II that all tested methods use linear search for finding the image with the lowest distance, which is unacceptable for searching large image collections (millions of images) or video collections.

REFERENCES

[1] Annotated groundtruth database. Department of Computer Science and Engineering, University of Washington. Retrieved on February 13th, 2013 from: http://www.cs.washington.edu/research/imagedatabase/groundtruth/.

[2] Savvas A Chatzichristofis and Yiannis S Boutalis. FCTH: Fuzzy color and texture histogram – A low level feature for accurate image retrieval. In *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS'08. Ninth International Workshop on*, pages 191–196. IEEE, 2008.

[3] Savvas A Chatzichristofis, Konstantinos Zagoris, Yiannis S Boutalis, and Nikos Papamarkos. Accurate image retrieval based on compact composite descriptors and relevance feedback information. *International Journal of Pattern Recognition and Artificial Intelligence*, 24(02):207–244, 2010.

[4] Thomas Deselaers, Daniel Keysers, and Hermann Ney. Features for image retrieval: An experimental comparison. *Information Retrieval*, 11(2):77–107, 2008.

[5] K Konstantinidis, A Gasteratos, and I Andreadis. Image retrieval based on fuzzy color histogram processing. *Optics Communications*, 248(4):375–386, 2005.

[6] Vedran Ljubovic and Haris Supic. Issue of resource usage in content-based image retrieval algorithms. In *Telecommunications (BIHTEL), 2012 IX International Symposium on*, pages 1–5. IEEE, 2012.

[7] Vedran Ljubovic and Haris Supic. Comparative study of color histograms as global feature for image retrieval. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2013 36th International Convention on*, pages 1059–1063. IEEE, 2013.

[8] Mathias Lux and Savvas A Chatzichristofis. LIRE: Lucene Image REtrieval: An extensible Java CBIR library. In *Multimedia (MM' 08). Proceedings of the 16th ACM international conference on*, pages 1085–1088. ACM, 2008.

[9] Bangalore S Manjunath, J-R Ohm, Vinod V Vasudevan, and Akio Yamada. Color and texture descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):703–715, 2001.

[10] Henning Müller, Stéphane Marchand-Maillet, and Thierry Pun. The truth about Corel-evaluation in image retrieval. In *Image and Video Retrieval*, pages 38–49. Springer, 2002.

[11] Henning Müller, Wolfgang Müller, David McG Squire, Stéphane Marchand-Maillet, and Thierry Pun. Performance evaluation in content-based image retrieval: Overview and proposals. *Pattern Recognition Letters*, 22(5):593–601, 2001.

[12] Chong-Wah Ngo, Ting-Chuen Pong, and Roland T Chin. Exploiting image indexing techniques in DCT domain. *Pattern Recognition*, 34(9):1841–1851, 2001.

[13] Bing Pan, Helene Hembrooke, Thorsten Joachims, Lori Lorigo, Geri Gay, and Laura Granka. In Google we trust: Users' decisions on rank, position, and relevance. *Journal of Computer-Mediated Communication*, 12(3):801–823, 2007.

[14] Gerald Schaefer. CVPIC colour/shape histograms for compressed domain image retrieval. In *Pattern Recognition*, pages 424–431. Springer, 2004.

[15] Gerald Schaefer and Michal Stich. UCID: An uncompressed color image database. In *Proc. SPIE 5307. Storage and Retrieval Methods and Applications for Multimedia 2004*, pages 472–480. International Society for Optics and Photonics, 2003.

[16] Shamik Sural, Gang Qian, and Sakti Pramanik. Segmentation and histogram generation using the HSV color space for image retrieval. In *Image Processing. Proceedings of the 2002 International Conference on*, volume 2, pages II–589. IEEE, 2002.

[17] Michael J Swain and Dana H Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[18] Ellen M Voorhees and Donna Harman. Common evaluation measures. In *The Twentieth Text Retrieval Conference (TREC 2011), Proceedings*, pages 500–255, 2012.

[19] James Ze Wang, Jia Li, and Gio Wiederhold. SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(9):947–963, 2001.

[20] Ka-Man Wong, Lai-Man Po, and Kwok-Wai Cheung. A compact and efficient color descriptor for image retrieval. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 611–614. IEEE, 2007.

[21] Zhenhua Zhang, Wenhui Li, and Bo Li. An improving technique of color histogram in segmentation-based image retrieval. In *Information Assurance and Security (IAS '09). Fifth International Conference on*, volume 2, pages 381–384. IEEE, 2009.