Comparative Study of Color Histograms as Global Feature for Image Retrieval

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

Abstract- Color histograms are one of the earliest and best known image features used in Content-Based Image Retrieval (CBIR). There is a wealth of scientific work on this topic. However, different papers vary in the specific ways of determining histograms and distance between them. In this paper authors attempt to classify various types of histograms used in literature and compare them using contemporary datasets and metrics for evaluation. Histograms are compared based on their retrieval performance as well as resource usage.

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

I. Introduction

Color histograms are one of the oldest and best known global features used in image retrieval [1]. They are often used as a baseline to compare other retrieval methods with. Recent research suggests that global color histograms provide decent retrieval performance when compared to other retrieval techniques [2].

There is a wealth of scientific work on the topic of histograms. However, approaches to dimensionality reduction, color spaces, distance calculation etc. differ. There is a lack of studies with quantitative comparison of various methods for calculating color histograms and distance between them.

Use of common datasets with defined ground truth and standard metrics for evaluating various retrieval methods is an important issue in CBIR [3]. Such datasets and metrics have become available only relatively recently. Therefore, it is of interest to evaluate various methods based on histograms in this new context.

Another often neglected issue is that of resource usage. The progress of multimedia technology did not make that issue any less relevant today [4].

In this paper we will attempt to determine the optimal approach for calculating color histograms and their distance, from perspective of retrieval performance as well as resource usage. We hope to achieve this by performing a large number of query-by-example (QbE) requests on referential annotated datasets and calculate average retrieval performance. Our focus is on the use-cases of personal image collection and Internet image search engines.

II. Previous Work

Earliest work on using color histograms in image retrieval is [1]. It uses histograms in RGB color space. RGB model is convenient for image representation in computers because it matches the way images are displayed on CRT monitors. Other papers on color histograms also use RGB color model (e.g. [2]).

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. [5] [6] [7] [8] [9]). It is also used for the Scalable Color Descriptor (SCD) feature of MPEG-7 standard for multimedia content description [10]. A closely related color model is HSL (also called HSI).

Another model often used in CBIR is Y'UV color model, also known as $Y'C_BC_R$ [11] [12] [13] (not to be confused with CIE YUV model). Primary use of this model is in digital television and popular image formats such as JPEG, where it allows for efficient compression of color information. This suggests that histograms in YUV model could have high tolerance for quantization. Formula for direct conversion from RGB to YUV ($Y'C_BC_R$) is given in ITU-R BT.601 standard and used in JPEG images. Closely related models are YIQ [6] (used in older versions of NTSC format) and $Y'P_BP_R$.

None of the above color spaces is uniform. Papers [14] and [15] suggest that using a uniform color space such as CIE L*a*b* or Luv should deliver superior retrieval performance. However, conversion from RGB to Lab requires calculation of cube root, making it computationally expensive.

Second problem in histograms is that of dimensionality reduction. 3D histograms are impractical and use a lot of memory. Earliest work on histograms [1] uses the following approach: Each of color channels is first quantized by taking a number of most-significant bits. Then these bits are

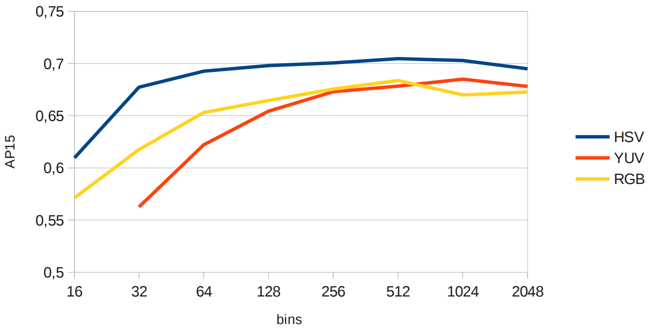

Figure 1: Relationship between number of bins and retrieval performance. Graph shows average precision (AP15) in Wang1000 dataset for histograms in HSV, YUV and RGB color spaces using optimal quantizations for each color space. Very similar results were observed in other datasets as well.

concatenated to form an index into a single histogram [6] [7]. In this paper we call such method *combined histograms*.

For example, a commonly used quantization scheme is HSV 16*4*4 [10]. 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 into a single histogram index I 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. Some papers (e.g. [8], [9]) use more complex formulas for quantization of each color channel.

Another approach is to calculate separate histogram for each of color channels. Those separate histograms are then concatenated into a single feature vector [5] [11] [16]. For example, if histograms in RGB color space use 32 bins of 1 byte each, total feature vector would have 96 bytes. We call this method *split histograms*.

Typically, normal (non-cumulative) histograms are used, although [5] argues for using cumulative histograms.

Thus obtained histograms will contain values in range [0,N] where N is the total number of pixels in image. In order to efficiently represent it in memory, such histogram needs to be normalized (divide all members with maximum) and quantized. It is typical to quantize each bin to one byte (8 bits) though lower quantizations may be preferable [10].

Also, many of the values in histograms are zeros, suggesting that a simple run-length coding could result in a more efficient storage [5].

Various transforms can be applied to further decrease the size of feature vector. A number of papers (e.g. [10]) use Haar transform. Reference [5] proposes storing only the first three moments of the histogram.

Finally, there must be a method for calculating distance between two feature vectors. Most papers use simple Manhattan distance (L1) or Euclidean distance (L2) [5] [11] [12] [13], although later work suggests that optimal distance metrics for histograms may be Jeffrey divergence (Jensen-Shannon divergence) [2] or histogram intersection [9] [15].

III. Methodology

Use of standardized, annotated datasets is an important issue in CBIR. [3] To this purpose we used four datasets consisting of photograph collections that are often cited in literature: Wang SIMPLYcity [18], UW [19], UCID [20], and MIRFLICKR-25000 [21].

A common metric for comparing retrieval methods is Precision-Recall graph. Two methods are compared by overlaying their graphs [21]. 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 [21] [22]. This is a standard measure used by NIST in its TREC contest.

AP15 - average value for precision in the first 15 retrieved images [21]. 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 [23].

Average Normalized Modified Retrieval Rank (ANMRR) is explained in [6] and [10]. This is the standard metric used in ImageCLEF competition.

Each image in dataset is annotated with a number of tags. A retrieved result is considered relevant if it has at least one tag in common with query image. We wanted to also gauge the

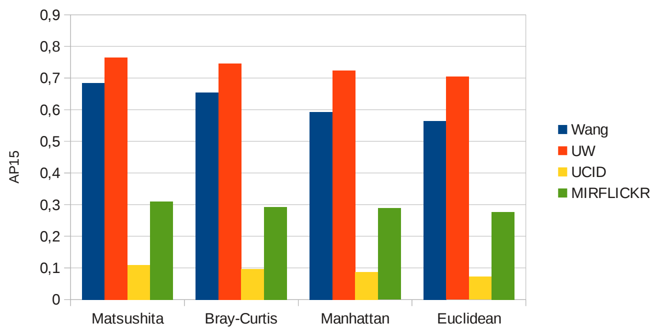

Figure 2: Comparison of distance metrics. Graph shows average precision (AP15) for RGB combined histogram with 512 bins. Each dataset is shown in different color. Similar results were observed with other color spaces.

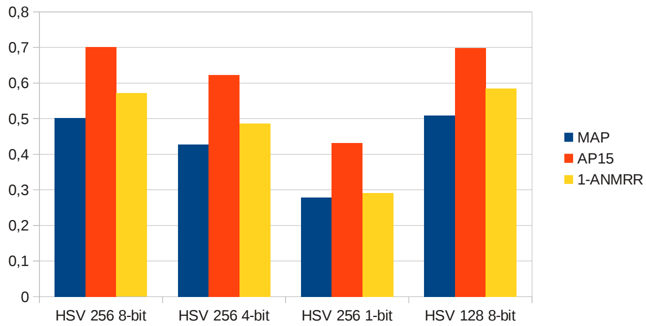

Figure 3: Effect of bin quantization on retrieval performance. Tests were performed on Wang1000 dataset with a combined histogram in HSV model, using histogram with 256 bins quantized to 8, 4 and 1 bit per bin. For comparison, histogram with 128 bins is given. Three retrieval metrics are depicted with different colors (since ANMRR gives scores where less means better retrieval, we display 1-ANMRR). Similar results were observed with other color spaces and datasets.

ability of tested methods to quickly retrieve results of higher relevance. 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 higher value corresponds to 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.

A large number of experiments were performed using open-source tool ETFImageSearch which can be downloaded from website: http://f.etf.unsa.ba/imagesearch Experimental results can be found at the same website (section Files).

When forming conclusions we looked for results where all metrics agree to a large extent across most datasets. We've found that there isn't a big difference between AP15 and AWP15. We've also found that results from AP15/AWP15 are more consistent across all four datasets as compared to MAP and ANMRR.

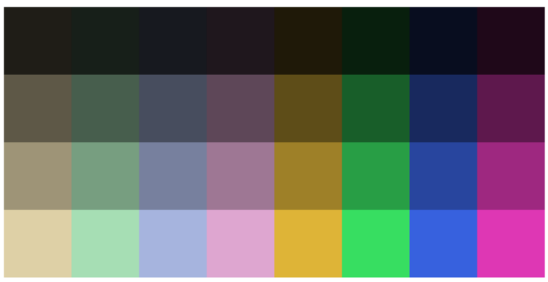

*Figure 4: Palette of colors obtained when HSV color space is quantized to 32 values using 4*2*4 scheme.*

IV. Experimental Results

Firstly, we attempted to determine the optimal **number of bins** for *combined histograms*. For simplicity, we tested histogram sizes that are a power of two.

Histograms with 256 and 512 bins on average give the best performance across all datasets. Histograms in HSV model also perform well with 128 bins. (Figure 1) Using a larger number of bins (1024 and more) gives inconsistent results. Results for 4096 and more bins are omitted from Figure 1 due to their inconsistent nature, but they can be found on previously mentioned project website. Therefore using such large histograms is not advised and generally there are no benefits above 4096 bins.

Histograms with less than 256 bins continue to give good results and only at below 64 bins there is a considerable decrease in retrieval performance.

Next, we made a comparison of **color models** used. In Wang, UW and UCID datasets the best results are obtained using HSV color model, followed by RGB and YUV color model. HSV also shows better tolerance of quantization and better performance with lower number of bins (Figure 1). Only in MIRFLICKR dataset HSV color model using combined histogram gives inexplicably poor results. This anomaly requires further research.

We've found no statistically relevant difference between YUV, YC_BC_R and YIQ models. For the rest of this paper only YUV color space will be considered. Results for HSV and HSL are similar, and decision on which is better is inconclusive. We've found that CIE Lab model doesn't give a sufficient boost in performance to warrant its much more complex histogram calculation.

Further we tested different **quantization schemes**. RGB space doesn't favor any particular channel when it comes to quantization. With Lab, Luv and YUV spaces greater quantization of the first (luminance) channel is slightly preferred. Thus optimal results in YUV model were obtained with 4*8*8 quantization giving 256 bins. In HSV and HSL color spaces smaller quantization of first channel (hue) is preferred. Optimal schemes were 32*4*4 (512 bins), 16*4*4 (256 bins, also used in [10]), 8*4*4 (128 bins), 4*4*4 (64 bins), 4*2*4 (32 bins).

Custom quantizations described in [8] and [9] give excellent results that sometimes exceed all other types of histograms. However, increase in complexity doesn't justify their use. Histograms where sizes per channel are a power of two can be calculated using binary shifting which is a very cheap operation.

Next issue tested was use of **split vs. combined** as well as **normal vs. cumulative** histograms. Split histograms are of interest because they result in a smaller feature vector. Cumulative split histograms appear to give very good results when Euclidean or Manhattan distance metric is used. However, when optimal distance metric is used for each given type of histogram (as outlined below), combined histograms consistently outperform normal split histograms, which in turn outperform cumulative split histograms.

This proves true even when a very large level of quantization is used. Combined histogram with 32 bins in HSV model (4*2*4) outperforms all kinds of split histograms in every metric, whereas the smallest split histogram with usable performance uses 8+8+8 bins.

Cumulative combined histograms display a sharp decrease in retrieval performance regardless of distance measure used.

An important issue revealed by our testing was that of **optimal distance metric** (Figure 2). 15 distance metrics described in [24] were tested with all described types of histograms. We've found that the optimal distance metric for all types of histograms is Matsushita distance. However this distance metric is fairly computationally intensive, since it requires calculation of square root per each bin. In applications where this is an issue, Bray-Curtis distance (and the very similar Soergel distance) provide excellent results.

With cumulative split histograms, best results are usually obtained using Euclidean and Manhattan distance, although results are inconsistent. Bray-Curtis as well as Bhattacharya distance sometimes give better results.

Finally, we explore the issue of **histogram quantization**. In our initial test, each histogram bin was represented as a 64-bit floating point number (double). When bins are quantized as 8-bit integers, there is no detectable loss in retrieval performance. However, further quantization gives a sharp decrease in performance (Figure 3). Histograms with 256 4-bit bins always perform below histograms with 128 8-bit bins. Quantization to 1 bit per bin, as suggested in [10], gives very poor performance.

Optimal distance metrics for 4-bit histograms remain Matsushita and Bray-Curtis/Soergel distance.

V. Conclusions

On average, the best retrieval score across all datasets is obtained using combined histogram in HSV color space with 256 bins (using 16*4*4 color quantization scheme). Such histogram is also used as an image descriptor in MPEG-7 standard [10]. Each bin is then normalized and quantized to 8 bits (1 byte) per bin. Best distance calculation approach is Matsushita distance.

Described method gives a feature vector of 256 bytes per image. In certain applications this can be considered too large. We've found that the best approach to reduce feature vector size is to reduce the number of bins per histogram. Given feature vector of less than 64 bytes, the best retrieval performance is obtained with HSV combined histogram using 4*2*4 quantization and 8 bits per bin.

The only exception to this is found in MIRFLICKR dataset where histograms in YUV color space perform better than HSV and HSL. This anomaly requires further research.

We've found that using split or cumulative histograms is not beneficial. Most papers arguing for their use are making a comparison based on Euclidean or Manhattan distance, which is a suboptimal distance metric for combined histograms. When best distance metric is used for each type of histogram, combined and non-cumulative histograms always outperform split and cumulative histograms.

This conclusion can be easily understood since combined histograms give a statistic of human-perceptible colors in an image, while split histograms observe each channel separately. For example, HSV combined histogram with 32 bins gives a count of common named colors such as orange, pink, brown etc. (Figure 4) while this information is lost in a split histogram. The problem of histogram "shifting" described in [5] is resolved by using a better distance metric. Further research in the area of distance metrics for histograms could yield even better results and is a promising target for futher research.

References

[1] M.J. Swain, D.H. Ballard, "Color Indexing", *International Journal of Computer Vision*, 1991;7(1), pp. 11-32.

[2] Thomas Deselaers, Daniel Keysers, Hermann Ney. "Features for image retrieval: an experimental comparison", *Information Retrieval*, 11.2 (2008), pp. 77-107.

[3] Henning Müller, Stéphane Marchand-Maillet, Thierry Pun, "The truth about corel-evaluation in image retrieval", *Image and Video Retrieval* (2002), pp. 38-49.

[4] Vedran Ljubovic, Haris Supic, "Issue of resource usage in content-based image retrieval algorithms", *Proceedings of BIHTEL 2012*, Sarajevo, 2012.

[5] Markus Stricker, Markus Orengo, "Similarity of color images", *Proc. SPIE Storage and Retrieval for Image and Video Databases*, Vol. 2420, No. 381-392, 1995.

[6] Savvas A. Chatzichrisofis, Konstantinos Zagoris, Yiannis S. Boutalis, Nikos Papamarkos, "Accurate image retrieval based on compact

composite descriptors and relevance feedback information", *International Journal of Pattern Recognition and Artificial Intelligence*, 24, no. 02 (2010), pp. 207-244.

[7] Shamik Sural, Gang Qian, Sakti Pramanik. "Segmentation and histogram generation using the HSV color space for image retrieval.", *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 2, pp. II-589. IEEE, 2002.

[8] Zhenhua Zhang, Wenhui Li, Bo Li. "An improving technique of color histogram in segmentation-based image retrieval", *Information Assurance and Security, 2009. IAS'09. Fifth International Conference on*, vol. 2, pp. 381-384. IEEE, 2009.

[9] Chong-Wah Ngo, Ting-Chuen Pong, Roland T. Chin, "Exploiting image indexing techniques in DCT domain", *Pattern Recognition 34*, no. 9 (2001), pp. 1841-1851.

[10] Bangalore S. Manjunath, J-R. Ohm, Vinod V. Vasudevan, Akio Yamada, "Color and texture descriptors", *Circuits and Systems for Video Technology, IEEE Transactions on 11*, no. 6 (2001), pp. 703-715.

[11] Zhe Ming Lu, Su-Zhi Li, Hans Burkhardt, “A content-based image retrieval scheme in JPEG compressed domain”, *International Journal of Innovative Computing, Information and Control*, Vol 2, No 4, pp. 831-839, August 2006.

[12] Ramadass Sudhir, Lt Dr S. Santhosh Baboo, "An Efficient CBIR Technique with YUV Color Space and Texture Features", *Computer Engineering and Intelligent Systems* 2, no. 6 (2011), pp. 85-95.

[13] Tienwei Tsai, Yo-Ping Huang, Te-Wei Chiang, "Fast image retrieval using low frequency DCT coefficients", *Proceedings of the 10th conference on artificial intelligence and applications*, Kaohsiung, Taiwan. 2005.

[14] K. Konstantinidis, A. Gasteratos, I. Andreadis, "Image retrieval based on fuzzy color histogram processing", *Optics Commun.* 248(4-6) 15 (2005), pp. 375-386.

[16] Gerald Schaefer, "CVPIC colour/shape histograms for compressed domain image retrieval", *Pattern Recognition* (2004), pp. 424-431.

[17] A. Jain, A. Vailaya, "Image Retrieval using Color and Shape", *Pattern Recognition*, vol. 29, no. 8, pp. 1233-1244, 1996.

[18] James Z. Wang, Jia Li, Gio Wiederhold, “SIMPLIcity: Semantics-sensitive Integrated Matching for Picture Libraries”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 23, No 9, pp. 947-963, 2001.

[19] *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/

[20] George Schaefer, M. Stich, "UCID - An Uncompressed Colour Image Database", *Storage and Retrieval Methods and Applications for Multimedia 2004*, *Proceedings of SPIE*, Vol. 5307, pp. 472-480, 2004.

[21] M. J. Huiskes, M. S. Lew, "The MIR Flickr Retrieval Evaluation", *ACM International Conference on Multimedia Information Retrieval (MIR'08)*, Vancouver, Canada, 2008.

[22] Henning Müller, Wolfgang Müller, David McG Squire, Stéphane Marchand-Maillet, Thierry Pun, "Performance evaluation in content-based image retrieval: Overview and proposals", *Pattern Recognition Letters* 22, no. 5 (2001), pp. 593-601.

[23] "Appendix: Common Evaluation Measures", *The Twentieth Text REtrieval Conference (TREC 2011) Proceedings*, 2012.

[24] Daniel Ruby, "The Value of Google Result Positioning", Retrieved on February 13th, 2013, from: http://insights.chitika.com/2010/the-value-of-google-result-positioning/

[25] Maria Hatzigiorgaki, Athanassios N. Skodras. "Compressed domain image retrieval: a comparative study of similarity metrics", *Visual Communications and Image Processing*. 2003.