

An Empirical Study of Color-Spatial Retrieval Techniques for Large Image Databases

Chia Yeow Yee Kian-Lee Tan Tat Seng Chua Beng Chin Ooi
Dept. of Information Systems & Computer Science, National University of Singapore

Abstract

In this paper, we present an experimental evaluation of three color-spatial retrieval techniques — the signature-based technique, the partition-based algorithm and the cluster-based method. The experimental study is performed on an image database consisting of 12,000 images. With the proliferation of image retrieval mechanisms and the lack of extensive performance study, the experimental results can serve as guidelines in selecting a suitable technique and designing a new technique.

1. Introduction

In a color-spatial retrieval technique, the color information is integrated with the knowledge of the colors' spatial distribution to facilitate content-based image retrieval. However, existing work [2, 4, 5] have largely focused on the retrieval effectiveness of the proposed mechanisms, and have been done with little comparative study. This paper presents three color-spatial retrieval techniques – the partition-based approach, the signature-based algorithm and the cluster-based method. We implemented the techniques and compare their retrieval effectiveness (in terms of normalized recall and precision) and retrieval efficiency (in terms of the speed of retrieval). The latter is important since a slow system will not be useful or practical for large image databases. The experimental study is performed on an image database consisting of 12,000 images. Our results provide insights into the strengths and weaknesses of these techniques. To our knowledge, such empirical study is the first of its kind.

2. Integrated Color-Spatial Techniques

2.1. Partition-based Approach

In the partition-based color-spatial technique, the color-spatial information of an image is modeled by partitioning

the image into $k \times k$ equal-sized sub-areas, and the color information within each sub-area is represented by a color histogram. The degree of similarity between the corresponding j th histograms of two images Q and D, can be computed as follows:

$$S_h(Q, D, j) = \sum_{i=1}^m \left(1 - \frac{|NH_{i,j}(Q) - NH_{i,j}(D)|}{\max(NH_{i,j}(Q), NH_{i,j}(D))} \right)$$

where m is the number of colors supported by the system, $NH_{i,j}(IMG)$ is the normalized pixel count for color i of the j th histogram of image IMG. Therefore, the degree of similarity between two images Q and D is given as follows:

$$S(Q, D) = \frac{1}{k^2} \sum_{j=1}^{k^2} wt_j \times S_h(Q, D, j)$$

where wt_j is a weight factor assigned to the j th histogram. Note that $\sum_{j=1}^{k^2} wt_j = 1$. By assigning a higher weight to a histogram implies that the similarity value of that histogram contributes more to the overall similarity computation of the two images.

To facilitate speedy retrieval, we employ a B⁺-tree index whose keys are (partition,color) pair. In this way, by traversing the index, we can restrict the search space to only those images that have the same color in the same partition.

2.2. Signature-based Approach

The signature-based approach [1] partitions an image into a grid of $m \times n$ cells of equal size. For a given color, each cell is examined to determine the percentage of the total number of pixels in the cell having that color. If this percentage is greater than a pre-defined threshold value, then the cell is said to be represented by that color. Depending on the threshold value, a cell may have no color representative or it may have more than one representative. Given a color, a cell is either represented or not represented by it. As such, each cell can be represented by a bit — if the cell satisfies the threshold value, the bit is set; otherwise, it is cleared. Hence, for each color, a bitstream (called the *color*

signature) that captures the spatial distribution of that color is obtained.

Let Q_i and D_i denote the signatures of color i for a query image Q and a database image D respectively. Let the representative color sets of Q and D be C_Q and C_D respectively. Then, the similarity measure, $S(Q, D, i)$, between Q and D for a color $i \in C_Q$ can be determined as:

$$S(Q, D, i) = \sum_{j \in S_p} \frac{BS(Q_i \wedge D_j)}{BS(Q_i)} \times SM(i, j)$$

where $BS(B)$ denotes the number of bits in the bitstream B that are set, and \wedge represents the bitwise logical-AND operation. This computation also considers the contribution of the perceptually similar colors of color i . S_p is the set of colors that are perceptually similar to color i as derived from the color similarity matrix SM . $SM(i, j)$ denotes the (i, j) entry of matrix SM . The similarity measure between two images Q and D is then given by:

$$S(Q, D) = \sum_{\forall i \in C_Q} S(Q, D, i)$$

A weighted similarity measure that favors some colors over others can also be obtained:

$$S_{weight}(Q, D) = \sum_{i \in C_i} S(Q, D, i) + wt \times \sum_{i \in C_c} S(Q, D, i)$$

where C_i and C_c are the set of background and object colors of Q respectively, and $wt (> 1)$ is the weights given to the object colors. A weight greater than 1 can be assigned to the object colors to give a higher weight to images with similar object colors as that of the query image.

Given a query image, its set of color signatures are extracted, and compared against those stored in the database. Database images that are similar can then be ranked and retrieved accordingly.

2.3. Clustered-based Approach

The clustered-based approach is based on the observation that two images appear to be similar if both of them have large patches (*clusters*) of similar colors at roughly the same locations in the images. The approach extracts a set of k *dominant* colors of an image, half of which are taken from the center of the image (representing object colors) while the other half are obtained from the entire image (representing background colors). For each of the dominant colors, the sequential 8-connected component algorithm is applied on the image to obtain a set of clusters in the image space. Note that each cluster has only one color and is represented as a rectangle. These clusters are then ranked (regardless of color) in descending order of their sizes (area

of the rectangles). The k largest clusters will be picked as the dominant clusters to be used as the color-spatial information of the image.

Given two clusters, the amount of overlap would give a very good measure of the similarity between the two MBRs. Two rectangles overlap only if they have the same color, and they intersect in the image space; the degree of overlap is given by the number of pixels intersected. Therefore, the similarity measure is given as follows:

$$SIM(Q, D) = \sum_{k=1}^{N_c} \sum_{i=1}^{N_k^Q} \sum_{j=1}^{N_k^D} overlap(C_Q(i, k), C_D(j, k))$$

where N_c is the total number of colors in the representative set, N_k^{IMG} is the total number of clusters of color k in image IMG , and $overlap(C_Q(i, k), C_D(j, k))$ is the intersection between cluster $C_Q(i, k)$ and cluster $C_D(j, k)$.

The retrieval process for this scheme works as follows. The image database is initially preprocessed to determine the clusters (color-spatial information) of the images. Given a sample query image, its k clusters are first extracted. The color-spatial information of each image in the database is then compared with those of the query image using the similarity function described above. The images can then be ranked based on the percentage of overlap, retrieved and displayed in that order.

To speed up retrieval, we employ a multidimensional index to handle the spatial distribution of the clusters in the image. We choose to use the R-tree [3] structure, a multi-dimensional generalization of the B-tree, that preserves height-balance.

3. Performance Evaluation

A prototype image retrieval system is implemented on a SUN Sparc Workstation running Solaris 2.5 using C/C++ programming language with the user interface written in X/Motif. The image database holds the set of 12,000 images used for testing. The images, together with the associated text descriptions, are acquired commercially from Kagem Corporation. The images are divided into eighteen categories by the image supplier. Major categories include: animal, art, computer, food, nature, travel, etc.

All the three color-spatial techniques are implemented; and for each technique, the database images are preprocessed to extract the appropriate color-spatial information. The *retrieval module* of the prototype accepts sample images from the user as queries through a *graphical user interface* module. Depending on the technique to be employed, the color-spatial information of the query image is extracted, and the appropriate indexes and database is searched for matching images. Candidate images are ranked based on the similarity measure and presented to the

users. The graphical interface is designed to display the top 60 retrieved images ranked in decreasing order of similarity. As not all relevant images are retrieved, the commonly recall-precision metrics are not appropriate. Instead, we compute the normalized precision (P_{norm}) and normalized recall (R_{norm}) [6] for each query:

$$P_{norm} = 1 - \frac{\sum_{i=1}^R (\log \text{rank}_i - \log i)}{\log \frac{N!}{(N-R)!R!}}$$

and

$$R_{norm} = 1 - \frac{\sum_{i=1}^R (\text{rank}_i - i)}{(N - R)R}$$

where N is the number of images in the collection, R is the number of relevant images in the collection and rank_i is the rank order of a relevant image i as retrieved by the system. In our computation, we set N to 12,000. For the relevant images that are not ranked among the top 60, their ranks are set to near the mid way between 60 and 12,000. Assuming the mid way point between 60 and 12,000 is avg_rank , relevant images that are not ranked among the top 60 are ranked in our computation as avg_rank , $\text{avg_rank}+1$, $\text{avg_rank}+2$ and so on.

To study the effect of certain type of images on our system performance, we pick 10 images classified into four categories: (a) images with distinct center object, (b) images with no distinct center object, (c) images with object on the left and (d) images that do not fall into the above categories. The number of relevant images for each query is obtained using keyword search, followed by visual inspection of the images retrieved. Each image in the database has a list of keywords supplied by the supplier of the database and this list is found to be precise in describing the contents of the images.

3.1 Tuning the Algorithms

We begin by first tuning each of the algorithms for optimal performance. Due to space limitation, we shall not discuss the details here. For all experiments, we restrict the total number of colors in the system to be 11. This coarse representation is used for two reasons. First, it captures the color similarity. Second, our preliminary experiments show that having more colors does not improve the effectiveness but may be less efficient.

For the partition-based approach, we investigated the number of partitions, and the number of color keys to be employed for optimal performance. Our results showed that 9 partitions (i.e., partitioning the image to 3×3 partitions) and using 2 color keys provide good retrieval effectiveness and efficiency. Our results also showed that, in general, giving the center partition a higher weight can lead to better

performance as very often the object of interest is in the center of the image.

For the cluster-based technique, the number of clusters can affect performance. A small number of clusters can speed up the retrieval process and lower the storage cost at the expense of the retrieval accuracy. Adding more clusters may increase the retrieval accuracy since these clusters are likely to lead to more retrievals. However, having too large a number of clusters will eventually lead to poor performance. This is because the set will contain small clusters which may contribute to the retrieval of irrelevant images. Further, the storage of these large number of spatial rectangles, and retrieval and comparison cost are likely to become high too. We investigated the effect of the number of clusters on retrieval accuracy, and our results show that it is reasonably sufficient to use only 2 clusters (one from the background, and one from the center) to discriminate between the images.

Finally, for the signature-based method, the size of the grid is fixed at 32×32 giving a total of 1024 cells. Furthermore, we also study the choice of an appropriate threshold value to be used in determining whether a cell is represented by a color, and found a value of 40% to be effective. We also showed that a total of 2 dominant colors, 1 from the background and 1 from the object can lead to reasonably optimal performance.

3.2 Comparative Evaluation

This experiment compares the performance of the three methods. Table 1 shows the result on the retrieval effectiveness of the schemes. First, we note that not a single scheme performs the best in all cases. Instead, there are instances when each of the scheme is superior. For example, the signature-based scheme is good for images I_1 , I_2 and I_3 ; the partition-based scheme is effective for image I_4 ; the cluster-based method gives the best result for I_{10} . Second, we note that the signature-based scheme performs the best in terms of the average normalized precision and recall. Both the cluster-based and partition-based are equally good.

Table 1 compares the retrieval efficiency of the various schemes in terms of time (in sec). We shall first focus on the single-level results, which correspond to the indexing schemes described in Section 2. As can be seen, the signature-based approach is generally the most efficient. This is not surprising in view of its compact representation. However, as before, there are cases where the other two schemes can perform slightly faster, though the cluster-based approach is the least efficient overall.

Though the results of the single-level schemes are below 6 secs, we still do not consider the schemes satisfactory since they may not be able to scale well. In other words, if the collection of images increases to 120,000 or 1,200,000

Image	Retrieval Effectiveness						Retrieval Efficiency					
	Partition		Cluster		Signature		Single-level			Multi-level		
	Precision	Recall	Precision	Recall	Precision	Recall	Cl.	Part.	Sign.	Cl.	Part.	Sign.
I_1	0.5325	0.7268	0.5951	0.7573	0.7314	0.8384	6.0	2.0	3.0	2.0	0.54	0.30
I_2	0.5343	0.7169	0.6327	0.7876	0.8157	0.8889	5.0	6.0	3.0	1.0	0.45	0.28
I_3	0.2518	0.5644	0.5781	0.7471	0.8298	0.8989	6.0	3.0	3.0	1.0	0.43	0.31
I_4	0.4042	0.6584	0.1770	0.5315	0.1351	0.5061	5.0	9.0	3.0	1.0	0.59	0.31
I_5	0.1580	0.5189	0.1354	0.5061	0.1575	0.5189	3.0	2.0	3.0	1.0	0.45	0.33
I_6	0.2286	0.5570	0.1823	0.5316	0.1761	0.5315	5.0	4.0	3.0	1.0	0.40	0.30
I_7	0.3141	0.6094	0.1584	0.5226	0.4345	0.6677	2.0	5.0	3.0	1.0	0.33	0.27
I_8	0.4192	0.6709	0.4377	0.6712	0.4512	0.6837	4.0	2.0	3.0	1.0	0.51	0.33
I_9	0.3111	0.6094	0.3265	0.6239	0.3601	0.6384	6.0	7.0	3.0	2.0	0.54	0.30
I_{10}	0.3759	0.6708	0.6850	0.8480	0.4510	0.6716	5.0	5.0	3.0	2.0	0.45	0.34
Average	0.3530	0.6303	0.3908	0.6527	0.4542	0.6844	4.0	4.6	3.0	1.5	0.46	0.31

Table 1. Comparative evaluation.

images, the performance may degrade drastically. As a first cut, we enhanced the various schemes into multi-level structures. In our study, we restrict to two-level structures:

- The clusters under cluster-based scheme are organized into partitions with each partition containing clusters of a single color.
- The signature scheme is also enhanced in a similar fashion as the cluster-based approach: signatures are organized into partitions of the same colors.
- In the partition-based approach, each partition is further split into subpartitions based on the number of colors.

In this way, during the retrieval process, only the respective (sub)partition needs to be searched. Note that reorganizing the indexing structure does not affect the retrieval effectiveness.

We reevaluated the schemes under the multi-level structures. As shown in Table 1, the relative performance of the various schemes remains largely the same: the signature-based method is the most efficient, followed by the partition-based scheme and then the cluster-based technique. However, we noted a significant reduction in terms of the retrieval time in all schemes.

4 Conclusion

In this paper, we have presented an evaluation of three color-spatial image retrieval techniques — signature-based technique, partition-based algorithm and cluster-based method. Our results provided insights into the strengths and weaknesses of these techniques, which can

help in designing color-spatial techniques for large image databases.

Acknowledgement

This work is partially supported by the university research grants RP920640 and RP950658.

References

- [1] T. S. Chua, K. L. Tan, and B. C. Ooi. Fast signature-based color-spatial image retrieval. In *Proc. of the International Conference on Multimedia Computing and Systems'97*. June 1997.
- [2] Y. Gong, H. Chua, and X. Guo. Image indexing and retrieval based on color histogram. In *Proceedings of the 2nd International Conference on Multimedia Modeling*, pages 115–126, Singapore, November 1995.
- [3] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD Conference*, pages 47–57, May 1984.
- [4] W. Hsu, T. Chua, and H. Pung. An integrated color-spatial approach to content-based image retrieval. In *Proceedings of the 1995 ACM Multimedia Conference*, pages 305–313, San Francisco, CA, November 1995.
- [5] H. Lu, B. Ooi, and K. Tan. Efficient image retrieval by color contents. In *Proceedings of the 1994 International Conference on Applications of Databases*, pages 95–108, Vadstena, Sweden, June 1994.
- [6] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.