



Color Mining of Images Based on Clustering*

Lukasz Kobyliński and Krzysztof Walczak

Institute of Computer Science, Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665 Warszawa, Poland

Abstract. The increasing size of multimedia databases and the ease of accessing them by a large number of users through the Internet carries a problem of efficient and semantically adequate querying of such content. A metadatabase may be used to shorten query resolution time by trying to limit the number of images being thoroughly analyzed to a smaller subset, having a high probability of finding the query image. In the article we propose a simple but fast and effective method of indexing such image metadatabases. The index is created by describing the images according to their color characteristics, with compact feature vectors, that represent typical color distributions. We present experiment results of typical search schemes by querying the metadatabase index created using a few different approaches.

Keywords: image retrieval, image clustering, multimedia indexing, data mining

1 Introduction

Querying of images by content is an important part of any multimedia mining system and it has been thoroughly studied in the literature. As the datasets available simultaneously to thousands of people in the world grow, a problem of not only accurate but also very efficient methods arises. It is now not only a question how to find an image most probably similar to the given example, but also how to perform this query in the global network, in enormous number of databases containing enormous numbers of images.

A possible approach to this problem is to divide it to smaller subproblems of indexing the individual databases and introduce a global metaindex, which enables to perform a preliminary selection between the databases, while resolving a given query. In the case of answering a query by example, the features of the query image are firstly calculated and the metaindex is queried to receive a list of databases most probably containing images similar to the one given. Then, the databases are queried in the given order to find the particular images, most similar to the example. This way, having an efficient method of indexing the databases and querying the resulting metaindex, the cost of performing a more thorough search may be reduced significantly by reducing the problem domain.

* The research has been partially supported by grant No 3 T11C 002 29 received from Polish Ministry of Education and Science.

In this paper we propose to use the Binary Thresholded Histogram (BTH) to calculate the image features while creating the metadatabase index. It is a color mining based approach to performing image queries, which has a very low cost of computation and adequate effectiveness for distinguishing databases having varied content. The BTH was created having the most compact representation of an image's color features in mind, while retaining enough information to be sufficient for differentiating of images. It then seems to be a very good candidate as a basis for creating the metadatabase index, enabling to perform very efficient, preliminary search in a set of databases.

The rest of the paper is organized as follows: in Section 2 we present previous work regarding the creation and indexing of metadatabases and content-based image retrieval. In Section 3 we describe the database setup that will be used in further discussion. In Section 4 we give the details of our approach to image representation, which is then used in query processing. In Section 5 we describe the employed method of processing a given query. Section 6 presents experimental results of database selection and Section 7 closes with a conclusion and discussion on possible enhancements.

2 Previous Work

The concept of employing a metadatabase index to facilitate querying of image databases has been proposed in [2]. The authors presented the use of a metadatabase to efficiently select the most appropriate databases while processing a query. Two selection strategies have been proposed, which maximize the probability of finding an image similar to a template associated with the query image.

The idea of Binary Thresholded Histogram has been presented in [6]. It is an efficient method of color-based retrieving of images, which uses 40 very compact, less than 32-bits long color feature vectors to find similarities between images.

Recent image mining and image retrieval surveys, relevant to the discussed subject, include [3] and [5]. Our previous work towards data mining in image databases, where the representation of images is based on color and texture features of regularly partitioned photographs has been presented in [4].

3 Database organization

There are fundamentally two different cases where a metadatabase index can be applied. Whenever a data can be thought of as a set of databases—individual collections of objects, which can be independently queried, these collections may contain semantically distinct or mixed content. In the first case objects of one semantic type exist only in one of the databases, whereas in the second they may appear in any number of datasets. On the other hand, a single database may consist of objects of only one semantic type or multiple types, without influencing the proposed method.

A video security monitoring system, consisting of multiple cameras pointed at different objects, each recording data separately, may be an example of the

first possible type of metadatabase application. Here, a problem may be stated as finding still images from any of the cameras, most similar to the given video frame. In this case only one of the databases containing photographs from distinct cameras should be thoroughly analyzed to find frames similar to the given query image, while the rest may be ignored. The aim of the metadatabase index is then to eliminate all irrelevant databases from the query and direct at the only one containing appropriate images.

In the web search engine research area there is a well known problem of routing queries to a distributed index of documents [1]. The problem, which may be given to illustrate the second possible case of data organization and the application of a metaindex structure, emerges from the fact, that distributed search engines need to have their index partitioned into multiple machines, using a method that guarantees the best possible load balancing across the servers. Many methods have been proposed for routing text search queries, but they can not be directly used in the case of performing image queries. Contrary to the first example, here the individual parts of the partitioned index concern subsets of images found on the Web by the search engine crawler and each of the subsets may contain images of different semantic types.

We have performed experiments with data simulating both stated possibilities, but the initial metaindex creation procedure is performed similarly, regardless of the type of data being processed. In our approach the metaindex consists of a very compact representation of each of the individual databases, created on the basis of color feature values of the images contained in the databases. A given database is firstly processed to calculate the features of each of the images and then clustered to obtain the set of the most significant feature values. The database representation consists of features of cluster centroids found in the database, along with the information of the number of images contained in each of the clusters and the mean and variance values of their similarity to the cluster centroids.

4 Image representation

We have used the Binary Thresholded Histogram approach, described in [6], to calculate and compare the color features of the images. The choice was made on the premise that a very efficient method, both in calculation and performing comparisons, must be used when creating the metadatabase index, to allow quick preliminary differentiation between available subsets of data.

4.1 Calculating the Binary Thresholded Histogram

As the BTH is calculated in the HSV color space, the first step in obtaining feature values is to convert a given image from RGB to HSV color representation. Secondly, the image is uniformly divided into rectangular blocks, by applying a $m \times n$ grid, where m is the number of rows and n is the number of columns. Next, color histograms of resulting blocks are calculated, with the chosen number

of bins set for each of the HSV components. This way each of the images is represented by $m \times n$ feature vectors consisting of $h \times s \times v$ values. Finally, the values are converted to binary by thresholding, with a given minimum $t\%$ threshold. Thus any color range appearing in the image subblock on more than $t\%$ of the pixels is denoted with a logical 1, while other colors are denoted with a 0. A logical 1 in the feature vector is meant to indicate that a particular color range is sufficiently represented in the image region to be noticed by a human. Consequently, the setting of the threshold value should be made on the basis of a human reception of the photographs and we have experimentally chosen a value of 4%. It is possible to tune the BTH to a given "quality ratio" by changing the representation bit length.

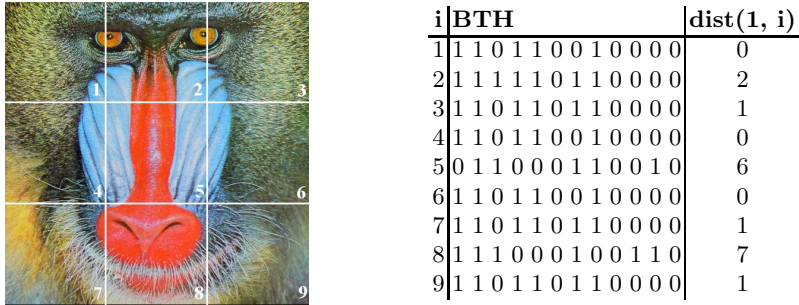


Fig. 1. A BTH representation of the *chimp* photograph and a few sample distance measure values between particular image tiles. The BTH threshold is set at $t = 4\%$ and $h = 4$, $s = 3$, $v = 1$ in this example.

This color feature representation approach is insensitive to small image translations, rotations, color shifts and brightness/contrast changes. The sensitivity depends on the set threshold value, chosen size of image tiles and the number of histogram color bins. For example in the case of translation, the representation will remain unchanged unless too many pixels are to be moved between neighboring tiles.

The binary representation allows for very quick comparisons, by performing a bitwise sum modulo 2 of any two feature vectors. Since each of the images is divided into $m \times n$ parts, a comparison of two images is accomplished by summing $m \times n \times h \times s \times v$ bits describing them. Formula 1 presents the calculation of the distance function between two images, for which histograms $x^{1\dots mn}$ and $y^{1\dots mn}$ are given.

$$d_{\text{BTH}}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{mn} \sum_{j=1}^{hsv} (x_j^i \oplus y_j^i). \quad (1)$$

Based on the distance, we can introduce a normalized similarity function between images \mathbf{x} and \mathbf{y} : $\text{sim}(\mathbf{x}, \mathbf{y}) = 1 - \frac{d_{\text{BTH}}(\mathbf{x}, \mathbf{y})}{mnhsv}$

Figure 1 presents an example of a BTH representation of an image divided into nine parts. A simple assessment of the method may be performed by comparing the distance measure values between image tiles (given as the rightmost table column) and the visual similarity of the tiles.

4.2 Database clustering

Since the image representation has a form of a binary vector, an adequate clustering algorithm must be employed to process such data, taking into account its relatively high dimensionality and computation time constraints. We have chosen to compare the CLOPE algorithm [7], which has been developed for effective clustering of transactional data and the well-known k-Means method with different distance measures.

The CLOPE algorithm relies on a concept of calculating simple parameters of cluster histograms and finding such clusterings, which optimize these parameters. This is accomplished by maximizing a global criterion function:

$$\text{Profit}(\mathbf{C}, r) = \frac{\sum_{i=1}^k \frac{S(C_i)}{W(C_i)^r} |C_i|}{\sum_{i=1}^k |C_i|}, \quad (2)$$

where $\mathbf{C} = \{C_1, \dots, C_k\}$ is a given clustering, $S(C)$ is the sum of item occurrences and $W(C)$ is the number of distinct items in the cluster C . The number of transactions in a cluster is denoted by $|C|$. By defining $S(C)$ as the size, $W(C)$ as width and $H(C) = S(C)/W(C)$ as the height of a cluster C , we see that $H(C)/W(C) = S(C)/W(C)^2$. The idea of the algorithm is then to maximize the „height” to „width” ratio of the clusters’ histograms, with a given input parameter r called repulsion, which allows to change the level of intra-cluster similarity.

Apart from the CLOPE algorithm we have also performed experiments with the k-Means clustering method, using several different distance measures, namely the BTH distance given by Formula 1, the city-block distance and the Euclidean distance. The city-block distance, given by the formula:

$$d_M(\mathbf{h}, \mathbf{g}) = \sum_{i=0}^{N-1} |h[i] - g[i]| \quad (3)$$

is normally equal to the distance measure proposed for BTH. The results are only different when we allow for non-binary values in the feature vector in intermediate calculations, as is the case in the vector averaging step performed in k-Means algorithm. We use only logical 0’s and 1’s in the case of BTH distance measure, which uses a bitwise exclusive-OR operation.

4.3 Cluster description

A few statistical parameters are calculated in the clustered database, to roughly assess the similarity distributions of images in the databases to the cluster centroids. A mean and variance of similarities to a given centroid are calculated and,

along with centroid's feature values and the number of images in the cluster, are included in the metadatabase index.

$$\text{mean}(db, t, l_t) = \frac{\sum_{i \in db \wedge \text{sim}(i,t) > l_t} \text{sim}(i, t)}{\text{num}(db, t, l_t)}, \quad (4)$$

$$\text{var}(db, t, l_t) = \frac{\sum_{i \in db \wedge \text{sim}(i,t) > l_t} (\text{sim}(i, t) - \text{mean}(db, t, l_t))^2}{\text{num}(db, t, l_t) - 1}, \quad (5)$$

where $\text{sim}(i, j)$ is a similarity measure between images i and j , $0 \leq \text{sim}(i, j) \leq 1$ and $\text{num}(db, t, l_t)$ is the number of images in database db with similarity greater than l_t to cluster centroid t .

Figure 2 presents the overall flow of database processing and the content of metadatabase index. At first the BTH feature values of the images in each of the databases are calculated. Having the representation in the form of binary vectors, a selected method of clustering is employed to form a chosen number of clusters. The feature values of cluster centroids are then used as the description of respective databases. The size of each cluster and the mean and variance of similarities between cluster content and cluster centroids are also included in the index.

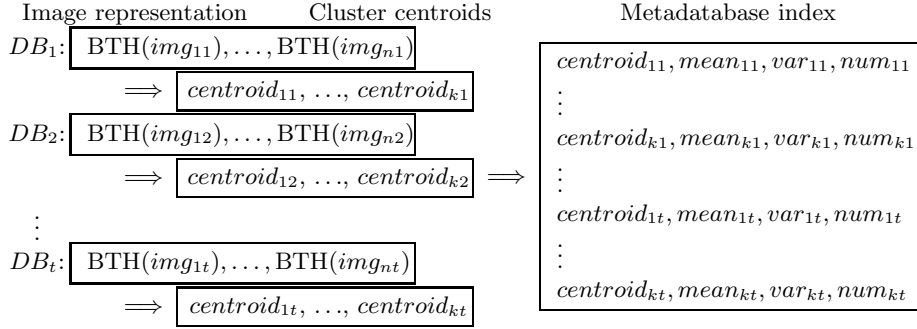


Fig. 2. An illustration of database processing and metadatabase index creation. The index is built on the basis of t databases, each containing a number of images belonging to different (or the same) semantic categories, for which BTH representation is calculated. The clustering step results in k cluster centroids for each of the databases. The index contains $k \times t$ centroid feature values, sizes of each cluster, as well as mean and variance values of distance measures of cluster elements to respective centroids.

5 Query execution

In a typical query by example scenario, a given image is firstly analyzed to calculate its features. This step is performed similarly, as described earlier in the case of database indexing. The feature values are then used to find the most

similar cluster centroids in the metadatabase index and then the corresponding databases, which satisfy certain conditions, depending on an employed database selection approach. Two such methods have been proposed in [2], of which we have used the mean-based approach in our experiments. The set D_i of relevant databases for the query q_i is chosen on the following criteria:

$$D_i = \{db \in DB \mid \text{num}(db, t, l_t) \geq v \wedge [x_t^{db}, y_t^{db}] \subseteq [a_{q_i, t}, b_{q_i, t}]\}, \quad (6)$$

where DB is the set of all databases, $[a_{q_i, t}, b_{q_i, t}]$ is a chosen similarity range of the query q_i to centroid cluster t within which the search is being conducted, $[x_t^{db}, y_t^{db}]$ is a calculated confidence interval, within which the mean similarity of images to centroid t is estimated to fall with a 99 percent chance and v is the minimum number of images similar to t in a database db for the database to be chosen.

The similarity range is defined by applying a given offset δ_q to the similarity measure value, such that:

$$a_{q_i, t} = \text{sim}(q_i, t) - \delta_q, \quad (7)$$

$$b_{q_i, t} = \text{sim}(q_i, t) + \delta_q. \quad (8)$$

The estimation of similarity mean value in a database db is performed by calculating a confidence interval at the 99% level ($\lambda = 0.01$): $x_t^{db} = \text{mean}(db, t, l_t) - E$, $y_t^{db} = \text{mean}(db, t, l_t) + E$, where:

$$E = z_{\lambda/2} \sqrt{\frac{\text{var}(db, t, l_t)}{\text{num}(db, t, l_t)}}. \quad (9)$$

6 Experimental Results

Given a set of ca. one thousand images of such categories as mountains, trees, flowers, people and buildings we have generated two series of datasets, one with images of a single original category included only in one of the possible databases and the second with the categories mixed randomly between the databases. The number of generated databases has been limited by the number of categories present in the test set.

The experiments have been conducted by taking one of the images from the set randomly and performing a query by example. In the case of applying the method to a set with original categories in only one of the databases, the quality of selection can be measured directly - by comparing the category label of the query image and the category of the selected database. In the second case of having a mixed content in each of the databases the quality is measured by precision and recall factors. This allows to assess the accuracy of the returned list of databases most probably containing images similar to the given sample. By denoting the number of relevant databases by A , the number of retrieved databases by B and the number of retrieved relevant databases by C we calculate precision as $P = C/B$ and recall as $R = C/A$.

The following parameter values have been set when performing the experiments: images partitioned into 3×3 blocks ($m = n = 3$), BHT with 64 color bins ($h = 8, s = 4, v = 2$), BHT threshold $t = 4\%$, similarity threshold $l_t = 0.5$, minimum number of similar images per database $v = 10$.

Table 1 presents a comparison of performance of the clustering methods during the preliminary database indexing phase. The results of clustering using the k-Means method are the averages of ten repeated experiments, as there is a random factor involved in the algorithm when selecting the starting cluster centers. The aim of the clustering is to find representative subgroups in each of the databases, from which distribution parameters are to be extracted and used in the metadatabase index. The clusters should thus be „pure” – ideally containing only one category of images. To assess the clustering performance we have used an entropy measure, calculated for each of the created clusters:

$$E_i = \sum_j -\frac{c(i, j)}{n_i} \log \frac{c(i, j)}{n_i}, \quad (10)$$

where $c(i, j)$ is the number of images in cluster i having a class label j and n_i is the number of all images in cluster i . The overall clustering entropy is calculated as the weighted sum of the individual cluster entropies: $E = \sum_i E_i n_i / \sum_i n_i$.

The CLOPE algorithm is an efficient clustering method, but not as accurate in performing BTH image representation clustering as the k-Means algorithm. It creates clusters with high mean similarity of the feature vectors, but also higher variance, which results in higher overall entropy of the clustering. The higher entropy of clustering using the BTH distance over the city-block distance metric is caused by loss of information while rounding non-integer values to perform the exclusive-or operation.

Table 1. Clustering quality of a mixed dataset with images of a) 3, b) 6 and c) 9 semantic categories

clustering method	number of clusters	similarity		clustering entropy
		mean	variance	
a) CLOPE ($r = 1.2$)	4	0.9528	0.047222	1.0627
k-Means, BTH distance	3	0.9177	0.006013	0.6593
k-Means, Euclidean distance	3	0.9174	0.008626	0.5690
k-Means, city-block distance	3	0.9172	0.006264	0.5579
b) CLOPE ($r = 1.3$)	7	0.9473	0.039157	1.6706
k-Means, BTH distance	6	0.9263	0.009208	1.0513
k-Means, Euclidean distance	6	0.9256	0.010810	0.8342
k-Means, city-block distance	6	0.9220	0.007342	0.8853
c) CLOPE ($r = 1.6$)	9	0.9675	0.039062	1.8656
k-Means, BTH distance	9	0.9291	0.012842	1.2935
k-Means, Euclidean distance	9	0.9268	0.015042	0.8890
k-Means, city-block distance	9	0.9208	0.007643	1.1133

The results of the query by example queries are presented in Table 2 and Table 3. In the case of databases with uniform content the purpose of the system is to return a ranked list of possible matches for specified query. To illustrate the performance of the proposed method two parameters have been included in the comparison of results for multiple similarity ranges (δ_q values, see Formula 7 and 8) of queries. The m_r value is the mean rank in the sorted list of results returned by the algorithm of the database that contains images of the same category label as the query image. Ideally, this value should be equal to 1 (correct database returned as the first in the list) to eliminate all irrelevant databases from further search. The p value, given as a percentage of all queries, is the part of queries that didn't match any database in the metaindex. The accuracy of selecting an appropriate database proved to increase when increasing the number of subdivisions of the databases from $k = 3$ to $k = 6$ clusters. This results in increasing the amount of information available in the metaindex and facilitates the query resolution.

Table 2. Metadatabase query by example results. Uniform databases case. Clustering with a) $k = 3$ clusters ($r = 1.2$ for CLOPE) and b) $k = 6$ clusters ($r = 1.5$).

clustering method	$\delta_q = 0.04$		$\delta_q = 0.05$		$\delta_q = 0.06$		$\delta_q = 0.07$		$\delta_q = 0.08$	
	m_r	$p\%$	m_r	$p\%$	m_r	$p\%$	m_r	$p\%$	m_r	$p\%$
a) CLOPE	2.88	21.71	3.74	7.34	3.66	1.94	3.62	2.88	3.88	-
BTH	2.23	9.00	2.48	6.48	2.3	0.98	3.54	-	3.13	-
Euclidean	2.40	16.53	2.90	9.01	2.51	0.98	3.03	-	4.12	-
City-block	2.65	10.62	2.46	1.94	2.81	2.88	3.31	-	3.75	-
b) CLOPE	2.67	38.42	3.02	32.22	3.45	10.62	3.59	5.61	3.48	1.94
BTH	2.06	26.81	1.95	12.17	2.27	4.72	2.48	0.98	2.69	-
Euclidean	2.45	17.89	2.20	12.93	2.48	4.72	2.84	1.94	2.74	-
City-block	1.96	12.93	1.98	-	2.27	-	2.11	-	2.71	-

The mean results of queries performed in the databases of a mixed content are presented as precision and recall values for three different δ_q values. While narrowing the similarity ranges of the queries results in a better precision value, the recall indicates that at most two-thirds of relevant databases are returned. In most applications maximizing the recall value is a priority and it may be achieved by increasing δ_q to at least 0.08 with only a small loss of precision.

7 Conclusions

In this article we have proposed an application of Binary Thresholded Histogram (BTH), a color feature description method, to the creation of a metadatabase index of multiple image databases and conducted experiments comparing several clustering approaches for this application. The BTH, despite being a very rough and compact representation of image colors, proved to be an adequate method

Table 3. Metadatabase query by example results. Mixed databases case. Clustering with a) $k = 3$ clusters ($r = 1.2$ for CLOPE) and b) $k = 6$ clusters ($r = 1.3$).

clustering method	$\delta_q = 0.04$		$\delta_q = 0.06$		$\delta_q = 0.08$	
	precision	recall	precision	recall	precision	recall
a) CLOPE	0.6157	0.4201	0.6456	0.6145	0.6376	0.8408
BTH	0.6160	0.6057	0.6117	0.8877	0.5936	0.9793
Euclidean	0.5892	0.6731	0.5694	0.9237	0.5990	0.9933
City-block	0.6369	0.6425	0.5883	0.8943	0.5912	0.9843
b) CLOPE	0.5671	0.5319	0.5730	0.7348	0.5902	0.9256
BTH	0.6257	0.5208	0.6271	0.7913	0.6122	0.9713
Euclidean	0.6287	0.4798	0.6367	0.8418	0.5924	0.9600
City-block	0.6543	0.5671	0.6210	0.8073	0.6047	0.9627

of describing the characteristics of image databases and creating a metadatabase index for querying large amounts of data. While the approach does not provide very high precision in returning the set of relevant databases, it still significantly reduces the final query execution time by limiting the number of databases to analyze. The parameters may be set to achieve 100% recall value, which guarantees that all relevant databases will be retrieved for each query.

References

1. Baeza-Yates R., Castillo C., Junqueira F., Plachouras V., Silvestri F.: *Challenges on Distributed Information Retrieval*, International Conference on Data Engineering (ICDE). Istanbul, Turkey, April 2007. IEEE CS Press.
2. Chang W., Sheikholeslami G., Wang J., Zhang A.: *Data Resource Selection in Distributed Visual Information Systems*, IEEE Transactions on Knowledge and Data Engineering, 10 (6): 926–946, 2001.
3. Datta R., Joshi D., Li J., Wang J. Z.: *Image Retrieval: Ideas, Influences, and Trends of the New Age*. ACM Computing Surveys, 2007.
4. Kobyliński L., Walczak K.: *Image Classification with Customized Associative Classifiers*, Proceedings of the International Multiconference on Computer Science and Information Technology. November 6–10, 2006. Wisła, Poland.
5. Lew M., Sebe N., Djeraba C., Jain R.: *Content-based Multimedia Information Retrieval: State-of-the-art and Challenges*, ACM Transactions on Multimedia Computing, Communication, and Applications, Vol. 2, No. 1, pp. 1–19, February 2006.
6. Walczak K.: *Image Retrieval Using Spatial Color Information*, Computer Analysis of Images and Patterns: 9th International Conference, CAIP 2001 Warsaw, 53–60.
7. Yang Y., Guan X., You J.: *CLOPE: a fast and effective clustering algorithm for transactional data*, ACM SIGKDD '02. July 23–26, 2002, Edmonton, Alberta, Canada.