THE METRIC HISTOGRAM: A NEW AND EFFICIENT APPROACH FOR CONTENT-BASED IMAGE RETRIEVAL

Agma J. M. Traina, Caetano Traina Jr., Josiane M. Bueno
Computer Science Department
University of Sao Paulo at Sao Carlos - Brazil
[agma | caetano | josiane]@icmc.sc.usp.br

Paulo M. Azevedo-Marques
Science of Image and Medical Physics Center of the Medical School of Ribeirao Preto
University of Sao Paulo at Ribeirao Preto - Brazil
pmarques@fmrp.usp.br

Abstract This paper presents the *metric histogram*, a new and efficient technique to capture the brightness feature of images, allowing faster retrieval of images based on their content. Histograms provide a fast way to chop down large subsets of images, but are difficult to be indexed in existing data access methods. The proposed metric histograms reduce the dimensionality of the feature vectors leading to faster and more flexible indexing and retrieval processes. A new metric distance function *DM*( ) to measure the dissimilarity between images through their metric histograms is also presented. This paper shows the improvements obtained using the metric histograms over the traditional ones, through experiments for answering similarity queries over two databases containing respectively 500 and 4,247 magnetic resonance medical images. The experiments performed showed that metric histograms are more than 10 times faster than the traditional approach of using histograms and keep the same recovering capacity.

Keywords: content-based image retrieval, feature-based indexing techniques, histograms.

1. INTRODUCTION

The main practice used to accelerate the data retrieval in database management systems is indexing the data through an access method tailored to the domain of the data. The data domains stored in traditional databases - such as numbers and small character strings - have the total ordering

X. Zhou et al. (eds.), *Visual and Multimedia Information Management*

property. This property leaded to the development of the current Database Management Systems - DBMS. However, when the data do not have this property, the traditional access methods cannot be used. Data embedded in multi-dimensional domains are examples of information that cannot be directly sorted. Spatial access methods have been developed for such data domains, e.g., the R-tree [1] and its derivatives, and the methods derived from the k-d-tree [2].

Complex data, such images, video, sound, time series and DNA strings, among others do not have implicit order property. That is, these data cannot be sorted using only their raw information, and there is no direct way to create an access method to improve their retrieval over sequentially scanning all of them. In this paper we focus on images. Sequential scanning over a large set of images is impractical due to the high computational cost of comparing two images. There are two usual approaches to search for images in an image database. The first one attaches textual description to each image, and the search process finds the required images through these texts. This is not a good solution, as besides it cannot be automatized, the retrieval depends on the objective of the operator when he or she was describing the image, not on the objective of the query. The second approach uses image processing algorithms to automatically extract characteristics from the images, which are then indexed. This approach is independent of human subjectivity, and is the one adopted in this paper.

Indexing the extracted characteristics is usually done through spatial access methods, where each extracted feature is a dimension. Regarding image histograms, each bin is indexed as a dimension. However, the spatial access methods are efficient up to a limited number of dimensions, in general not larger than 10 or 20. Indexing images through their histograms (usually with 256 bins or more) is highly inefficient. It has been shown [3] that the main factor affecting the efficiency of a multi-dimensional access method is the intrinsic dimensionality of the data set, that is, the number of uncorrelated dimensions. However, this indexing could be done more efficiently if the correlations between close bins were used. Many attempts have been done to find more concise representations of histograms using dimensionality reduction techniques [4] [5]. All these attempts leads to histograms representations with a reduced number of dimensions, but with a pre-defined number of dimensions.

In this paper we consider that histograms of two images can have different numbers of correlations between its bins, so reduced histograms from those images could have distinct number of dimensions. This points out that reduced histograms are in a not spatial domain - this domain does not have a defined number of dimensions, as each reduced histogram has a different number of "reduced bins". However, as we show in this paper, it is possible to define a metric dissimilarity function between any pair of reduced histograms, so this domain turns out to be a metric one. A new class of access methods, applicable to metric domains, has been recently developed. So, our proposed method of histogram reduction leads to a metric representation of the histogram, that can be indexed through a metric access method [6]. As we will show, this approach leads to a very precise retrieval of data, whereas it can be up to 10 times faster than using traditional histograms, even when using the same access method.

The main motivation of this paper is to develop an efficient technique for image retrieval aiming medical applications. The work shown herein is part of the development of a picture archiving and communication systems (PACS) [7], used to integrate the information regarding patients in a hospital. A PACS is a valuable tool helping the physicians when diagnosing. The images of medical exams are stored together with conventional data (text and numbers). Therefore, it is possible to ask both queries regarding the content of images and the usual queries based on textual information.

The remaining of this paper is structured as follows. In the next section, we first give a brief history of spatial and metric access methods, including a concise description of the main techniques for extracting image features aiming their comparison by content. Section 3 introduces metric histograms as well as the *DM* distance function to be used for comparing the new proposed metric histograms. Section 4 presents the experiments performed in order to evaluate the proposed method, regarding precision/recall and time measurements. Section 5 presents the conclusion of this paper.

2. BACKGROUND

Image database management systems rely on efficient access methods to deal with both traditional data (texts, numbers and dates) and image data. The design of efficient access methods has attracted researchers for more than three decades. There are many consolidated access methods for traditional data, such as the B-tree and hash-based indexing access methods. Spatial data have been addressed through the so called Spatial Access Methods - SAMs, which include treelike structures, such as the R-tree [1], R+-tree [8], R*-tree [9], k-d-B-Tree[10], space- filling curves [11], spatial hash files [12], etc. An excellent survey on SAMs is given in [13]. Spatial domains have been widely used to store feature sets from images and other complex data.

The majority of features extracted from images can be seen as multidimensional points in a *n*-dimensional space. This is the case of histograms, moment invariants, Fourier features, and principal component analysis among others. The well-known spatial access methods downgrade the retrieval of objects when the dimension of the objects increases and a sequential scan processing would outperform such methods. Therefore, other access methods which can deal with high-dimensional data sets could be used in order to answer questions on such objects. The X-tree [14] and TV-tree [15] were developed to manage high dimensional data. However, it is necessary to highlight that in some occasions it is not possible to have all the obtained feature vectors with the same number of components (dimensions). Thus, it is not possible to use any SAM. For such situations the Metric Access Methods - MAMs - were developed. In metric domains the only information available is the objects and the distances between them. The distance function is usually provided by a domain expert, who gathers the particularities of the target domain in order to compare objects.

Formally, given three objects, x, y and z pertaining to the domain of objects S, a distance function $d(\ )$ is said to be metric if it satisfies the following three properties:

i. symmetry: $d(x,y) = d(y,x)$,

ii. non-negativity: $0 < d(x,y) < 4,\ x\ \ y$ *and* $d(x,x) = 0$,

iii. triangle inequality: $d(x,y)\ \#d(x,z) + d(z,y)$

A metric distance function is the foundation to build MAMs, which were developed since the pioneering work of Burkhard and Keller [16]. They are now achieving an efficiency level good enough to be used in practical situations, as is the case of the M-tree [17], the Slim-tree [18] and the Omni-family members [19].

Data in metric domains are retrieved using similarity queries. The two most frequently used similarity queries are defined following.

- ***k-nearest neighbor*** or ***k-NN query***: $kNN{=}{<}s_q, k{>}$, which asks for the k objects that are the closest to a given query object center s_q, with s_q $0S$ (the object domain). For example, in an image database domain, a typical query could be: "*find the 5 nearest images to* $image_1$".

- ***range query***: $Rq{=}{<}s_q, r_q{>}$, which searches for all the objects whose distance to the query object center s_q is less or equal to the query radius r_q. Using the example previously given, a query could be: "*find all the images that are within 10 units of distance from* $image_1$".

Calculating distances between complex objects are usually expensive operations. Hence, minimizing these calculations is one of the most important aspects to obtain an efficient answer for the queries. MAMs minimize the number of distance calculations taking advantage of the triangular inequality property of metric distance functions, which allows to prune parts of the tree during the answering process. Metric access methods built on the image features have been successfully used to index images and are suited to answer similarity queries [20].

Direct comparison between images can be very costly. Thus, a common approach is to extract features from the images. The main features used to compare images are: color, shape and texture, as well as the spatial relationship between the image objects [21] [22] [23]. Many work has been done in the field of content-based image retrieval aiming to speed up such comparisons [24] [21], as well as proposing new techniques for image comparison, using color histograms [25], shape [26] and texture [27]. The algorithms to extract shapes and textures are very expensive and dependent on the application domain, so it is better to leave them as the last step for separating the images, when the candidate response set was already reduced using the other features. The importance of color or brightness histograms is due to the simplicity of getting and comparing them, operations that are performed in linear time.

3. THE PROPOSED IDEA - METRIC HISTOGRAMS

Images are commonly represented as a set of elements (*pixels*) which are placed on a regular grid. The pixel values are obtained from a quantization process and correspond to the brightness level. Thus, formally an image can be represented by the following notation:

Definition 1: An **image** *A* is a function defined over a two-dimensional range G= $[0,x_0]$ x $[0,y_0]$ taking values on the set of possible brightness values V=$[0,v_0]$. That is, $A = \{(x,y,\ v(x,y))/\ (x,y)\ 0\ G$ and $v\ 0\ V\}$.

Definition 2: The **histogram** $H_A(z)$ of an image *A* provides the frequency of each brightness value *z* in the image. The histogram of an image with *t* brightness levels is represented by an array with *t* elements, called bins.

Definition 3: The **normalized histogram** $NH_A(z)$ of an image *A* provides the frequency of each brightness value *z* in the image, given in percentage. The normalized histogram of an image with *t* brightness levels is also represented by an array with *t* elements.

Obtaining the normalized histogram of images is not a costly operation. The normalized histogram is invariant to geometrical transformations as well as to linear contrast transformations. Normalized histograms allow comparisons of images of any size, so geometric transformations performed on the source images will give the same histogram.

Figure 1 shows an image obtained from magnetic resonance and its associated normalized histogram. This image has a spatial resolution of 512x512 pixels displayed in 256 brightness levels, thus its histogram has 256 bins. Indexing histograms like this one requires indexing arrays with 256 elements or, in terms of indexing structures, dimensions. The distance (or dissimilarity) between two histograms can be measured as the summation of difference between each bin, which is know as the Manhattan distance function or L_1 norm.

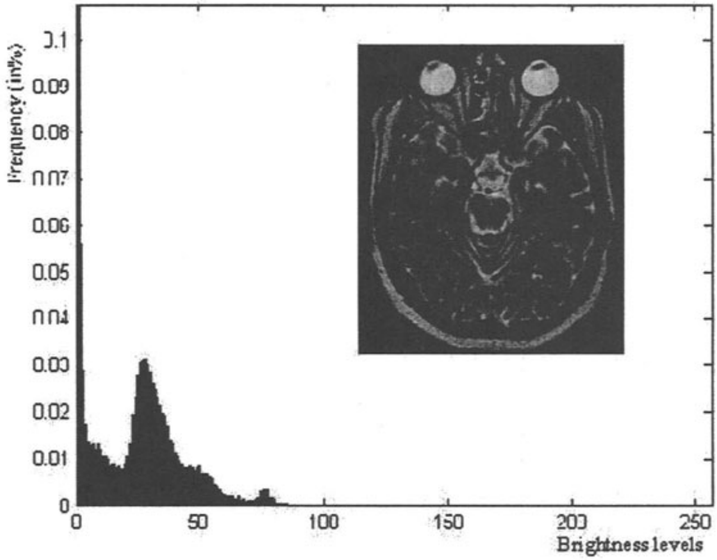


Figure 1 – An MR image and its normalized histogram.

In this work, we assume that the brightness levels are similar to its close levels, so the shape of the histogram can be kept using an approximation of it. Therefore, we propose to represent an approximation of a normalized histogram through a set of line segments. Histograms of different images can be approximated using different number of lines, so the approximation can be optimized to describe each histogram. Thus, these approximations are defined in a metric domain - this domain does not have a number of dimensions defined, as each approximation needs a different number of lines. This leads to the following definition.

Definition 4: A ***Metric histogram*** $MH_A(z)$ of an image A is defined as $M_H(A)=\{N_A, <b_k, h_k> \mid 0 \leq k < N_A\}$, which is a set of N_A buckets b_k, each one with the normalized height h_k.

A normalized histogram is composed by a number of bins. This number depends on the brightness resolution of the image, so it is a fixed number. In a metric histogram, the equivalent to the histogram bin is called a *bucket.* Each bucket corresponds to a line in the approximation. Buckets do not need to be regularly spaced. The number N_A of buckets in a metric histogram depends on the acceptable error in the approximation process and on the image itself. Each bucket k is a pair $<b_k, h_k>$, where b_k is the index of the rightmost bin of the original histogram represented in bucket k, and h_k is the normalized value of the rightmost bin represented in bucket k. Notice that b_0 is always zero. To simplify the notation, we indicate the bucket b_k of the metric histogram of image A as A_{bk}, and the normalized value h_k of the metric histogram of image A as A_{hk}. Figure 3(a) presents the metric histogram of two images A and B, and the components of their metric histograms.

The algorithm used to generate the metric histogram finds the ***control* points**, which are the inflection, minimum and maximum local points on the histogram, in order to create the approximation curve on the normalized histogram. Thus, the histograms are seen as functions in a 2-D space. Figure 2 shows the metric histogram obtained from the histogram in figure 1.

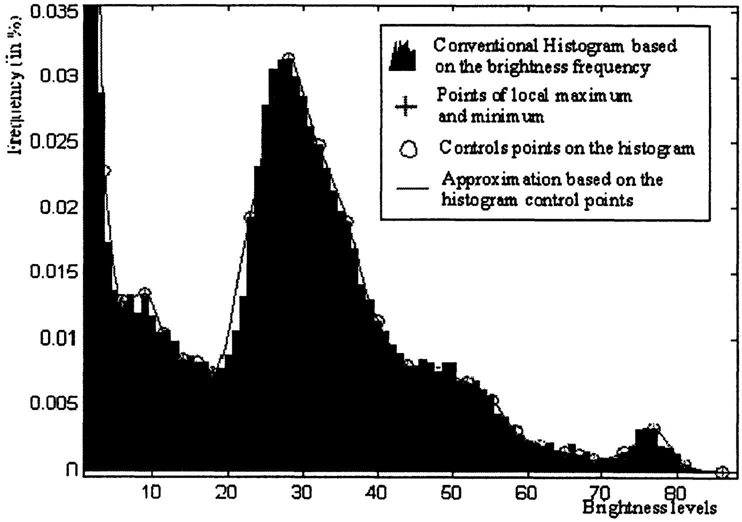


Figure 2 – Deriving a metric histogram through the approximation curve over the normalized histogram.

Regarding metric histograms, a question that arises is how to compare them, as the number of buckets and the spawning of the buckets from different histograms are variable. That is, the usual distances for histograms, such as Euclidean, Chi-square, Manhattan, Kolmogorov-Smirnov, Kuiper, between others [4] cannot be used. Therefore, we developed a new metric distance function in order to quantify the dissimilarity between metric histograms.

Definition 5: The distance function *MHD*( ) between two metric histograms is given by the non-overlapping area among the two curves representing the metric histograms, i.e.

$$MHD(M_H(A), M_H(B)) = \int_0^{end\_longest\_M_H} \left| M_H(A,x) - M_H(B,x) \right| dx$$

Figure 3 gives an example of how to calculate the distance between two metric histograms and figure 4 depicts the algorithm developed to calculate this distance function. In figure 3(a) the two histograms are overlapped, and the intersecting points and the ones which limit the buckets are shown. Figures 3(b) to 3(d) show how such points are used to calculate the area inside each region (in the steps of the algorithm in figure 3). Note that the number of steps is greater than or equal to the number of buckets of the histogram with more buckets. This is due to the fact that as the width of the buckets is variable, in some occasions they must be divided in order to obtain the area between the two metric histograms. This is exemplified in figure 3(b). Figures 3(c) and 3(d) shows the next two steps performed to calculate the *MHD*() function, helping to understand how the algorithm works. When one of the metric histograms finishes before the other one, that is, when N_A < N_B, the calculation of the distance also stops. It is straightforward to demonstrate that the distance *MHD*( ) is metric.

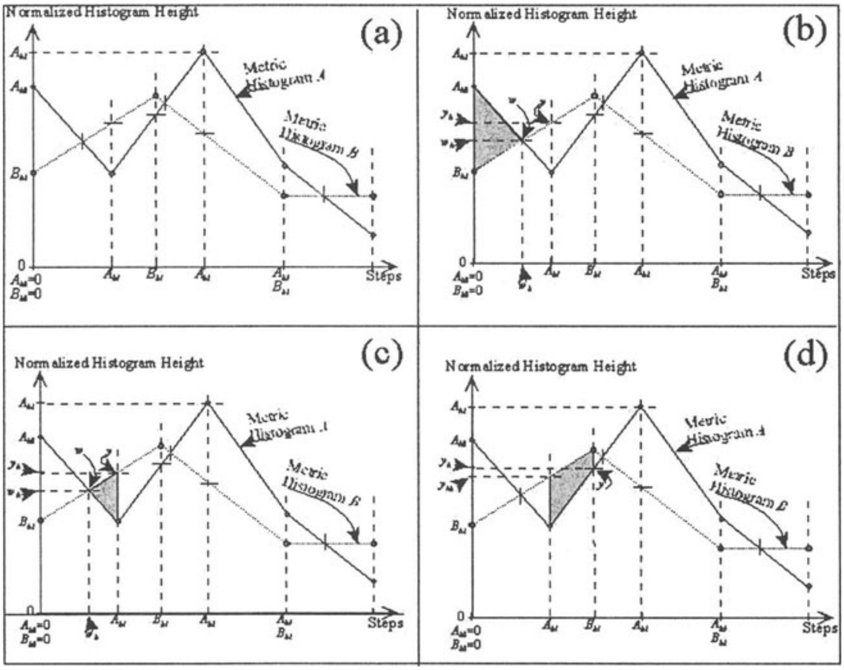


Figure 3 – Distance between two metric histograms through calculating the area between them. (a) Two metric histograms *A* and *B*, and the points used to specify the steps of the algorithm. (b) First step of the algorithm, exemplifying when the two MH intersects each other. (c) Second step of the algorithm. (d) Third step of the algorithm.

Ours experiments show that the number of buckets in the metric histogram is much smaller than the number of bins of normalized histograms, coming from 256 bins to 20 - 32 buckets in the metric histogram. It is important to recall that the metric histograms are obtained from normalized histograms. Thus, the following properties hold:

Property 1 - An original image and the same one scaled, translated or rotated will have the same metric histogram.

Property 2 - The metric histograms are curves in the space, thus the metric histograms can be adjusted at its beginning and ending. Therefore, metric histograms are also invariant to the image contrast.

```
The MHD distance function: calculating the distance between two metric histograms
Input: the two metric histograms $M_H(A)$ and $M_H(B)$
Output: the distance between $M_H(A)$ and $M_H(B)$.
  set $dist=0$, $s=0$, $a=0$, $h_a=A_{b0}$ , $h_b=B_{b0}$ , $i=1$ and $j=1$
  while there is more steps to compare
      if bucket $A_{bi} < B_{bj}$, then
              calculate the value of $B$ at position $A_{bi}$ as $y=(A_{bi}, y_2)$
              set $bm=A_{bi}$, $base=bm-s$, and $y_1=A_{hi}$
              increment $i$
      else
              calculate the value of $A$ at position $B_{bj}$ as $y=(B_{bj}, y_1)$
              set $bm=B_{bj}$, $base=bm-s$, and $y_2= B_{hj}$
              increment $j$
      if line $((a, h_a), (a, h_b))$ intersects line $((b_m, y_1), (b_m, y_2))$, then
              calculate the intersection $w=(w_b, w_h)$
              calculate $area_1= \left|(w_b - a) * \frac{h_a - h_b}{2}\right|$ and
              $area_2= \left|(b_m - w_b) * \frac{y_1 - y_2}{2}\right|$
              add $area_1 + area_2$ to $dist$
      else
              calculate $area_1 = base * \frac{h_a + y_1}{2}$ and
              $area_2 = base * \frac{h_b + y_2}{2}$
              add $|area_1 - area_2|$ to $dist$
      set $h_a=y_1$ and $h_b=y_2$
      set $a=base + a$
  return $dist$
```

Figure 4 – Algorithm for calculating the *MHD* distance function.

These two properties highlight that some restrictions of using histograms for image retrieval are overpassed using metric histograms. That is, it becomes invariant to geometric transformations (scale included) and contrast.

4. EXPERIMENTS

We implemented our proposed metric histogram and the distance *MHD*() using the Slim-tree as the metric access method to answer similarity queries. The Slim-tree has been used because it allows the minimization of the overlap between nodes in the structure as well to measure this overlap. So far, only the Slim-tree has these capabilities, which have been shown to be useful when dealing with multi-dimensional and non-dimensional datasets [28], as is our two classes of histograms.

We have used two image databases in the experiments. The first one, named*"MRHead500"*, has 500 human brain images obtained by magnetic resonance tomography (MRT), and the second one, named *"MRVarious4247"*, has 4,247 images from various human body parts also obtained by MRT. Each image has 256 brightness levels and different spatial resolution. We extracted the normalized histogram and generated the metric histogram of each image, creating two sets of features for each database. Each set of features was indexed using the Slim-tree structure (four Slim-trees in the total). The normalized histograms were compared using the Manhattan distance function over the 256-element arrays, and the metric histograms where compared using the *MHD* distance function with the number of buckets varying from 11 to 32.

The main objective of this paper was to define a faster way to pre-select images in a database, which is performed based on the image histograms, to reduce the need of image comparisons when answering similarity queries. To evaluate our proposed method, the experiments compared the set of images resulting from answering nearest-neighbor queries using normalized histograms and metric histograms as well. To this intent, we assumed that the results obtained from the normalized histograms, which is the traditional approach, are the correct ones. The objective was to compare the images retrieved using their metric histogram with the images retrieved using their normalized histogram. We calculated the *precision* and *recall* measurements on the results for each set of images obtained. Precision and recall are parameters commonly used to evaluate information retrieval systems as well as image retrieval systems [4] [29]. Recall indicates the proportion of relevant images in the database which has been retrieved when answering a query. Precision, by the other hand, is the proportion of the retrieved images that are relevant for the query. Formally, let X be the set of relevant images for a given query, Y be the set of retrieved images and x, y and z be the number of images in the sets $X \cap Y$, $\overline{X} \cap Y$ and $X \cap \overline{Y}$, respectively.

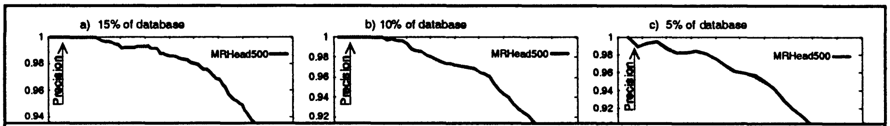
Thus, precision and recall can be expressed through the following conditional probabilities:

$$precision = P(X \mid Y) = \frac{P(X \cap Y)}{P(Y)} = \frac{x}{x+y}$$

$$recall = P(Y \mid X) = \frac{P(Y \cap X)}{P(X)} = \frac{x}{x+z}$$

4.1 Precision and recall measurements

Figures 5 and 6 present the precision and recall curves obtained when asking nearest-neighbor queries for six different numbers of neighbors over the *MRHead500* and *MRVarious4247* databases respectively. The numbers of neighbors refer to portions of the database, varying from 0.5% to 15% of each image database. That is, to ask *k*-NN queries requiring 1% of the database, means to ask NN queries with *k*=5 for the *MRHead500* database, and with *k*=43 for the *MRVarious4247*. The curves present the average value from asking 50 queries for each number of neighbors, with the center image randomly chosen from images in the database. In these experiments, each query is submitted to both Slim-trees, the one indexing the normalized



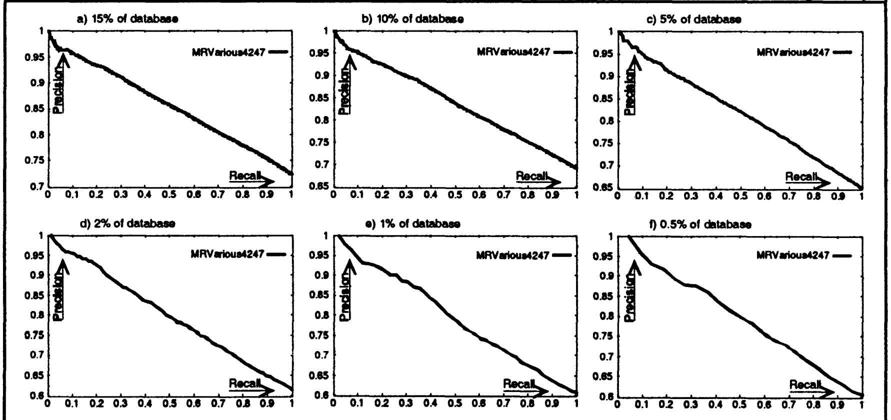


Figure 6 – MRVarious4247 database: Precision vs. Recall plots when answering *k*NN-queries where the number *k* is taken as a percentage of the database size, (a) 15%, (b) 10%, (c) 5%, (d) 2%, (e) 1%, (f) 0.5%.

histogram, and that indexing the metric histogram. Each answer set is sorted by the similarity of each histogram to that of the center of the query. The plots are calculated verifying that increasing numbers of histograms in the answer set of the tree indexed by the metric histogram is present in the answer set of the tree indexed by the normalized histogram. As it can be seen, the retrieval precision is high, always over 60% even for the *MRVarious4247* heterogeneous data set. With respect to the more homogeneous *MRHead500* data set, the results are always better than 70% of precision, even for queries retrieving 0.5% of the database with 100% of recall.

4.2 Timing comparison

Two questions tackled in this section are: (a) How much is the time difference for indexing the image databases through conventional and metric histograms? (b) Is the difference in time relevant when answering queries over the normalized and the metric histograms?

Table 1 reports the wall-clock times for building the Slim-tree in order to index the normalized (conventional) and the metric histograms. It can be seen that it is much faster to build the index over metric histograms (58% faster in the small data set, and 690% faster in the larger data set). The extra time needed to get the approximation over the conventional histograms in order to obtain the metric histograms pays off during the queries. The time needed to create the metric histograms is equivalent to the time to build the index tree using the metric histogram, but this is done only once and is small when comparing to the time to answer queries.

Database	**Histogram**	**Time** (in seconds)
MRHead500	Metric	4.62
	Conventional	7.31
MRVarious4247	Metric	33.76
	Conventional	266.79

Table 1 – Wall-clock time to build the Slim-tree using metric or normalized histograms from the *MRIHead500* and the *MRVárious4247* image data sets.

We also measured the number of distance calculations performed per second, and found 1,289,400 *MHD* distances per second and 269,430 Manhattan distances per second. That is, the calculation of one *MHD* distance is in average 4,7 times faster than one Manhattan distance over a pair of 256-element arrays.

The largest gain in time when using metric histograms is to answer queries. Figures 7(a) and 7(b) show the total times needed to answer 50 *k*-NN queries, when the value of *k* is specified corresponding to percentage of the database. Thus, the numbers are proportional to the database size and the results can be compared to different database sizes. Figure 7(a) shows the times to answer queries on the *MRHead500* database, and figure 7(b) shows the times to answer queries on the *MRVarious4247* database. As it can be seen, the gain ranges from 4 times faster for small *k* (0,5% of the database) to more than 10 times faster for larger portions of the database (15% of the database). All measurements were taken using an Intel Pentium II 450 MHz computer running Windows NT. The software was implemented in Borland C++ language.

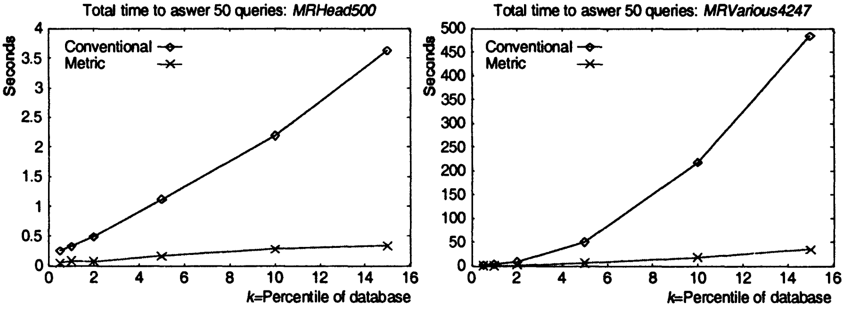


Figure 7 – Total time to answer 50 nearest-neighbor queries for each portion of the data set: 0.5, 1, 2, 5, 10 and 15%, using the conventional and the metric histograms. (a) *MRHead500* data set; (b) *MRVarious4247* data set.

5. CONCLUSIONS

Comparing two images is a very time consuming process. When queries are issued over an image database to retrieve information based on the image content, a filtering process takes place trying to reduce the number of images to be compared. This filtering operation uses features extracted from the images through relatively inexpensive algorithms. One of the features most frequently used in the early filtering stages is based on the image histograms.

The main objective of this paper is to define a faster way to execute the filtering process over an image database based on its histograms. To achieve this target we proposed a new technique, called the "metric histogram". We showed that using this technique the filtering step can be improved, achieving up to 10 times faster image selection for similarity queries. Moreover, the creation of index structures using metric histograms are at least 58% and up to 700% times faster as compared to structures created using the conventional histograms.

Metric histograms are defined in a metric domain, so the parameters that best describes one histogram can be used without compromising the whole set of images. To allow using metric access structures to answer similarity queries, we defined the *MHD*( ) distance function, which complies with the properties of a metric distance function. This function is also a faster way to compare histograms, as it can be executed in average 4,7 times faster than using the Manhattan distance function over 256-element vectors.

Metric histograms also have some very desirable side effects: it allows the retrieval of images in a way that is invariant to scale, rotation and translation of the objects in the image, and is also invariant to the contrast of the image. These effects enable that images similar to the target one, but with different contrast, scale, rotation, etc. can also be obtained without further computational effort.

ACKNOWLEDGMENTS

This research was supported, in part, by the scholarship grant from FAPESP (São Paulo State Research Foundation) number 98/03.209-6, and by CNPq (Brazilian National Research Council) under grants 52.1685/98-6, 860.068/00-7 and 35.0852/94-4. The authors also thank the HCRP (Clinical Hospital of Ribeirao Preto) for providing the data used in this work, and Humberto Razente as well as Fabio Chino for their support in the development of the software used in this work.

REFERENCES

[1] Guttman, A., "*R-Tree : A dynamic Index Structure for Spatial Searching*," in ACM SIGMOD, Boston, MA, 1984.

[2] Bentley, J.L., "*Multidimensional Binary Search Trees Used for Associative Searching*," *Comunications of the ACM*, vol. 18, pp. 509-517, 1975.

[3] Korn, F., Pagel, B.-U., Faloutsos, C., "*On the 'Dimensionality Curse' and the 'Self-Similarity Blessing'*," *IEEE TKDE*, vol. 13, pp. 96-111, 2001.

[4] Brunelli, R. and Mich, O., "*On the Use of Histograms for Image Retrieval*," in IEEE Intl. Conf. on Multimedia Computing and Systems (ICMCS), Florence, Italy, 1999.

[5] Berman, A. and Shapiro, L.G., "*Selecting Good Keys for Triangle-Inequality-Based Pruning Algorithms*," in Intl. Workshop on Content-Based Access of Image and Video Databases (CAIVD '98), Bombay, India, 1998.

[6] Chávez, E., Navarro, G., Baeza-Yates, R., Marroquín, J.L., "*Searching in Metric Spaces*," *to appear in the ACM Computing Surveys*, 2001.

[7] Siegel, E.L. and Kolodner, R.M., *Filmless Radiology*. New York City, NY: Springer Verlag, 1999.

[8] Sellis, T.K., Roussopoulos, N., Faloutsos, C., "*The R+-tree: A Dynamic Index for Multi-dimensional Objects*," in VLDB, Brighton, England, 1987.

[9] Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B., "*The R*-tree: An Efficient and Robust Access Method for Points and Rectangles*," in ACM SIGMOD, 1990.

[10] Robinson, J.T., "*The K-D-B-Tree: A Search Structure For Large Multidimensional Dynamic Indexes*," in ACM International Conference on Management of Data - SIGMOD, SIGMOD Conference 1981:, 1981.

[11] Liao, S., Lopez, M.A., Leutenegger, S.T., "*High Dimensional Similarity Search With Space Filling Curves*," in IEEE ICDE, Heidelberg, Germany, 2001.

[12] Samet, H., "Spatial data structures in Modern Database Systems: The Object Model, Interoperability, and Beyond,",, W. Kim, Ed.: Addison-Wesley/ACM Press, 1995, pp. 361-385.

[13] Gaede, V. and Günther, O., "*Multidimensional Access Methods,*" *ACM Computing Surveys*, vol. 30, pp. 170-231, 1998.

[14] Berchtold, S., Keim, D.A., Kriegel, H.-P., "*The X-tree: An Index Structure for High-dimensional data,*" in VLDB, Bombay, India, 1996.

[15] Lin, K.-I.D., Jagadish, H.V., Faloutsos, C., "*The TV-Tree: An Index Structure for High-Dimensional Data,*" *VLDB Journal*, vol. 3, pp. 517-542, 1994.

[16] Burkhard, W.A. and Keller, R.M., "*Some Approaches to Best-Match File Searching,*" *CACM*, vol. 16, pp. 230-236, 1973.

[17] Ciaccia, P., Patella, M., Zezula, P., "*M-tree: An efficient access method for similarity search in metric spaces,*" in VLDB, Athens, Greece, 1997.

[18] Traina, C., Jr., Traina, A.J.M., Seeger, B., Faloutsos, C., "*Slim-Trees: High Performance Metric Trees Minimizing Overlap Between Nodes,*" in Intl. Conf. on Extending Database Technology, Konstanz, Germany, 2000.

[19] Santos, R.F., Filho, Traina, A.J.M., Traina, C., Jr., Faloutsos, C., "*Similarity Search without Tears: The OMNI Family of All-purpose Access Methods,*" in IEEE ICDE, Heidelberg, Germany, 2001.

[20] Bozkaya, T. and Özsoyoglu, Z.M., "*Indexing Large Metric Spaces for Similarity Search Queries,*" *ACM Transactions on Database Systems (TODS)*, vol. 24, pp. 361-404, 1999.

[21] Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R., "*Content-Based Image Retrieval at the End of the Early Years,*" *IEEE Trans. on PAMI*, vol. 22, 2000.

[22] Aslandogan, Y.A. and Yu, C.T., "*Techniques and Systems for Image and Video Retrieval,*" *IEEE TKDE*, vol. 11, pp. 56-63, 1999.

[23] Güntzer, U., Balke, W.-T., Kiessling, W., "*Optimizing Multi-Feature Queries for Image Databases,*" in VLDB, Cairo - Egypt, 2000.

[24] Flickner, M. and alli, e., "*Query by Image and Video Content: The QBIC System,*" *IEEE Computer*, vol. 28, pp. 23-32, 1995.

[25] Pass, G., Zabih, R., Miller, J., "*Comparing Images Using Color Coherence Vector,*" in ACM Multimedia, Boston, MA, 1996.

[26] Tuytelaars, T. and Gool, L.v., "*Content-based Image Retrieval Based on Local Affinely Invariant Regions,*" in Information and Information Systems (Visual'99), 1999.

[27] Gimel'farb, G.L. and Jain, A.K., "*On Retrieving Textured Images from an Image Database,*" *Pattern Recognition*, vol. 29, pp. 1,461-1,483, 1996.

[28] Traina, C., Jr., Traina, A.J.M., Faloutsos, C., Seeger, B., "*Fast Indexing and Visualization of Metric Datasets Using Slim-trees,*" *IEEE TKDE*, vol. to appear, 2002.

[29] Yamamoto, H., Iwasa, H., Yokoya, N., Takemura, H., "*Content-based Similarity Retrieval of Images Based on Spatial Color Distributions,*" in 10th Intl. Conference on Image Analysis and Processing, 1999.