IMAGE RETRIEVAL WITH BINARY HAMMING DISTANCE

Jérôme Landré
CReSTIC, Université de Reims-Champagne-Ardenne
I.U.T., 9 rue de Québec, 10 000, Troyes, France

Frédéric Truchetet
Le2i, Université de Bourgogne
I.U.T., 12 rue de la Fonderie, 71 200, Le Creusot, France

Keywords: Content-based image retrieval, binary signature, multiresolution analysis.

Abstract: This article proposes a content-based indexing and retrieval (CBIR) system based on query-by-visual-example using hierarchical binary signatures. Binary signatures are obtained through a described binarization process of classical features (color, texture and shape). The Hamming binary distance (based on binary XOR operation) is used for computing distances. This technique was tested on a real natural image collection containing 10 000 images and on a virtual collection of one million images. Results are very good both in terms of speed and accuracy allowing near real-time image retrieval in very large image collections.

1 INTRODUCTION

Searching in large image collections is a challenging task for computer vision researchers. Internet and recent imaging technologies have facilitated the availability of private and public image collections leading to a need for efficient image searching tools.

Content-based image retrieval (CBIR) consists in working with images only without any other information. Images are too big to be used directly for indexing and retrieval, features extraction gives a feature vector per image which is a reduced representation of the image visual content.

Classical image features are mainly divided into three different families: color, texture and shape. In the proposed method, a binary feature extraction method gives a binary representation of feature vectors: binary signatures.

To compute distances between images, Hamming distance based on logical exclusive-or (XOR) function is used because it ensures great performances in terms of speed and accuracy.

This article is organized as follows. Section 2 describes related work on binary signatures for content-based image retrieval. In section 3, the proposed architecture is explained in depth. Section 4 defines the binary metric for comparing binary signatures. Experimental results are given in section 5. In section 6, a conclusion and several tracks to explore for future work are presented.

2 RELATED WORK

During the last decade, many image retrieval papers have been published. Getting fast and efficient CBIR systems is an interesting challenge because even with last generation processors, researchers have often to choose between speed and accuracy. To ensure optimized performances, distance computation must be rapid (Jacobs et al., 1995).

Several binary image retrieval techniques are based on binary coding of feature vectors. Color-based image retrieval with binary signatures (Nascimento and Chitkara, 2002) gave good results. Binary histograms have also been proposed (Kunttu et al., 2003). These methods give good results but work only with one family of feature: color.

The Hamming distance evaluates the number of bits that differ from two binary vectors. Fuzzy Hamming distance (Ionescu and Ralescu, 2005) has been published to solve Hamming distance limitations on real numbers. This distance is not used in this work because only binary signatures are computed, not real numbers.

Landré J. and Truchetet F. (2007).
IMAGE RETRIEVAL WITH BINARY HAMMING DISTANCE.
In *Proceedings of the Second International Conference on Computer Vision Theory and Applications - IU/MTSV*, pages 237-240

In our approach, users can work with color, texture and shape hierarchically to refine retrieval. These three families of features are not mixed together because they are independant. For example if a user wants to find "red cars" in a collection, color and shape have to be used. Texture will not be useful in this case. When you work with only one feature vector where the three features are mixed, useless features influence the final decision while they are not supposed to.

More and more methods are based on offline classification of feature vectors to build a visual search tree to browse the collection online. In our system, a query-by-visual-example method (Boujemaa et al., 2003) is used because time computing limitation is not really important in our retrieval process due to the high speed of binary computation.

3 PROPOSED ARCHITECTURE

Our system is based on binarization of classical features. There are two steps in the proposed system: offline and online. Let's consider an image collection $\mathcal{C}$ containing N images noted I_i where $i = 1..N$.

In the offline step (no user connected to the CBIR system), each image I_i of the collection $\mathcal{C}$ is transformed from RGB to Lab colorspace. Lab colorspace was chosen because distances computed in this space correspond to real perception of distances between colors. Then a multiresolution analysis (Calderbank et al., 1998) is computed at three resolution levels. Several classical features are extracted in color, texture and shape feature vectors. The binarization process is described further and leads to three binary signature per image: s_i^C, s_i^T and s_i^S.

The size of our signatures is 32-bits so that XOR operations can be processed into the microprocessor internal registers. Each bit in s_i^C, s_i^T and s_i^S represents a property which is true (1) or false (0). Thus each signature is a set of binary properties for the image I_i.

Figure 1 presents our query-by-example architecture. The binary extracted signature of the request image I_R is compared to every image I_i of the collection $\mathcal{C}$ and results are displayed on the user screen, sorted by increasing distance.

Features are organized into a 32-bits binary signature vector. For an image I_i, there are three binary signature vectors corresponding to color (s_i^C), texture (s_i^T) and shape (s_i^S). Bits in signatures represent the fact that the considered image satisfies a certain property or not.

- Color: Color properties are based on "a" and "b" maps values of "Lab" colorspace. There are 32 properties tested in every 32-bits color binary signatures. For instance, the first bit is to check property: — Does the mean value of "a" colormap at the coarser resolution is greater than 64 ? —. A value of 1 indicates this property is satisfied for this image, a value of 0 means it is not satisfied. So by associating several properties, our signature contains a checklist of color properties.

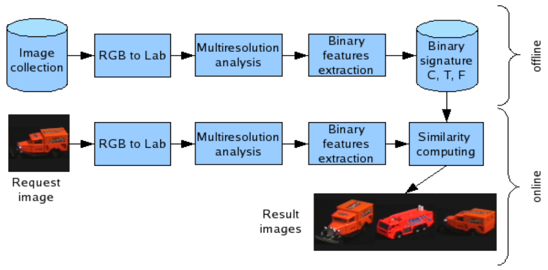

Figure 1: Architecture of the proposed system.

- Texture: Binary properties for texture are mainly based on the study of wavelets energy (square value of each coefficient) through the three different levels of resolution. For instance, the first bit is to check property: — Does the mean energy of "L" colormap for the coarser resolution is greater than 128 ? —.
- Shape: Shape properties are extracted from image contours of the "L" colormap (by a laplacian edge detector). For example, a typical property is: — Is there any continuous contour of the object longer than 30 pixels ? —.

So the entire process of binarization consists in transforming real world questions into binary answers. The underlying problem is the choice of properties.

Of course the list of properties is not exhaustive and any kind of question whose answer is yes (1) or not (0) is a potential binary property to use in our system. Once binary properties have been chosen, a similarity (or dissimilarity) metric must be used to compute distances between images, i.e. between signature vectors.

4 SIMILARITY COMPUTING

In order to evaluate distances between request image I_R and collection images I_i, a metric must be defined. We need a measurement method to tell how two binary signatures s_R (request) and s_i (i^{th} image in the collection) are similar (bit per bit). Therefore we want a similarity measure where the distance value will be the number of similar bits in the considered signa-

tures. Next table gives similarity truth table for the distance we want to define.
Considering the n^{th} bit of s_R and s_i, we want to know if they are similar or not:

$s_R[n]$	$s_i[n]$	$d(s_R[n], s_i[n])$	similarity
0	0	0	similar
0	1	1	not similar
1	0	1	not similar
1	1	0	similar

This truth table for needed similarity lead to a definition of similarity based on the XOR binary operator. The distance is computed as the number of bits whose value is 1 in the XOR result of the two given binary signatures. It is the definition of the Hamming distance.

For instance, let's consider two 8-bits signature vectors s_R^C and s_i^C. The distance between them will be $d_{\mathbb{I}} = \mathbb{I}(s_R^C \oplus s_i^C)$ where $\oplus$ is the XOR operator and $\mathbb{I}$ is the function that computes number of bits whose value is 1 in the binary XOR result.

Theorem 1 (Hamming) *$d_{\mathbb{I}}$ is a metric distance on $[0,1]^k$.*

By definition, the minimal and maximal distances $d_{\mathbb{I}}$ between two binary signatures in a k bits space($[0,1]^k$) are respectively 0 and k. Once the distance metric is defined, several experiments are possible to test it in real situation.

5 EXPERIMENTS

Several results using natural image collection are presented. This very well-known image collection contains 10 000 images. Experiments were performed on a Pentium 4 2GHz with 512 MB RAM laptop computer running Linux Fedora Core 5.

User interface was built upon web pages served by an Apache web server, with PHP for dynamic pages and MySQL for storage purpose. C programs using Intel IPP and OpenCV libraries were used for computing distances.

In order to measure efficiency of the proposed method, two parameters were studied: speed and accuracy.

Speed has been evaluated on the natural image database but also on a virtual set of one million random binary signature vectors to show real-time possibilities of the method. Computing times are given in seconds.

An image is represented by three 32-bits (4-bytes) signatures, s_i^C, s_i^T and s_i^S. The total image collection (N images) is represented by three arrays of unsigned int values whose length is N. So the total amount of memory needed to store our binary signature is $3 \times 4 \times N = 12 \times N$ bytes.

For the 10 000 images of natural image collection, the total amount of memory to store our signatures is $12 \times 10\,000 = 120\,000$ bytes. Computing time for distance is less than 10^{-3} second. So for a given request, distance $d_{\mathbb{I}}$ is computed real-time.

For the one million images virtual collection, the total amount of memory used is $12 \times 10^6 = 12$ Mb which is a small part of actual computer memory.

Table 1: Computing time for retrieval.

Collection (images)	$d_{\mathbb{I}}$ comp. time (sec.)
Natural image (10 000)	$< 10^{-3}$
Virtual (10^6)	$\simeq 0.59$

Results on table 1 show the computing time is very low leading to on-the-fly distance computing and to a real-time request-by-example retrieval system. Speed does not mean anything without accuracy.

Accuracy results are based on precision/recall plots for natural image collection.

Several request images were presented to the system. The result images for each request were sorted by increasing distance from the request leading to a precision and recall computation. This test process was applied on the full feature vector (containing color, texture and shape features) and on a hierachy of features (color then shape vectors). In the first case, only one distance had to be computed, in the second, one distance is computed for color features and another is computed for shape features.

Results are proposed on figure 2. This graph is the precision/recall graph based on a mean of twenty objects of the natural image collection. Results have been improved by using a hierarchy (color then shape) of binary signatures instead of one mixed (color+texture+shape) binary signature.

Examples about the advantage of using hierarchical features are proposed on figure 3. In this figure, using mixed features (color+texture+shape) gives bad results (false detection) compared to using color first then shape.

A comparison between mixed features and hierarchical features is shown on figure 3. The first image is the request image. If mixed feature vectors are used (a), many bad images are retrieved. If hierarchical feature vectors are used (b), the result is better with less mistakes than the previous case.

Two good examples of retrieval success with hierarchical feature vectors are given on figure 4. The