University of Alberta

Content-Based Image Retrieval Using Binary Signatures

by

Vishal Chitkara
Mario A. Nascimento
Curt Mastaller

Technical Report TR 00-18
September 2000
Revised April 2001

DEPARTMENT OF COMPUTING SCIENCE
University of Alberta
Edmonton, Alberta, Canada

Content-Based Image Retrieval Using Binary Signatures

Vishal Chitkara Mario A. Nascimento Curt Mastaller
Department of Computing Science
University of Alberta
Edmonton, Alberta T6G 2H1
Canada
{chitkara | mn | curt}@cs.ualberta.ca

Abstract

Significant research has focused on determining efficient methodologies for retrieving images in large image databases. In this paper, we propose two variations of a new image abstraction technique based on signature bit-strings and an appropriate similarity metric. The technique provides a compact representation of an image based on its color content and yields better retrieval effectiveness than classical techniques based on the images' global color histograms (GCHs) and color coherence vectors (CCVs). The technique is also well-suited for use in a grid-based approach, where information about the spatial locality of colors can be taken into account. Performance evaluation of image retrieval on a heterogeneous database of 20,000 images demonstrated that the proposed technique outperforms the use of GCHs by up to 50%, and the use of CCVs by up to 20% in terms of *retrieval effectiveness* – this relative advantage was also observed when using classical Precision vs Recall curves. Perhaps more importantly is the fact that our approach saves 75% of storage space when compared to using GCHs and 85.5% when using CCVs. That makes it possible to store and search an image database of reasonable size, e.g., hundreds of thousands of images, using a few megabytes of main memory, without the aid of (complex) disk-based access structures.
Keywords: Content-based image retrieval, CBIR, image databases, bit-string signatures, color histograms, grid-based CBIR.

1 Introduction

The enormous growth of image archives has significantly increased the demand for research efforts aimed at efficiently finding similar images within a large image database. One popular strategy of searching for images within an image database is called Query By Example (QBE) [dB99], in which the query is expressed as an image template or a sketch thereof, and is commonly used to pose queries in most Content-Based Image Retrieval (CBIR) systems such as IBM's QBIC[1], Virage's VIR Image Engine retrieval system[2], and IBM/NASA's Satellite Image Retrieval system[3].

Typically, the CBIR system extracts visual features from a given query image, which are then used for comparison with the features of other images stored in the database. The similarity function is thus based on the abstracted image content rather than the image itself. One should note that given the ever-increasing volume of image data that is available, the approach of relying on human-assisted annotations as a means of image abstraction is not feasible. The color distribution of an image is one such feature that is extensively utilized to compute the abstracted image content. It exhibits desired features such as low complexity for extraction, invariance to scaling and rotation, and partial occlusion [SB91]. In fact, it is common to use a *Global Color*

[1] `http://wwwqbic.almaden.ibm.com/`
[2] `http://www.virage.com/products/vir-irw.html`
[3] `http://maya.ctr.columbia.edu:8080/`

Histogram (GCH) to represent the distribution of colors within an image. Assuming an n-color model, a GCH is then an n-dimensional feature vector $(h_1,\ h_2,\ ...,\ h_n)$, where h_j represents the (usually) normalized percentage of color pixels in an image corresponding to each color element c_j. In this context, the retrieval of similar images is based on the similarity between their respective GCHs. A common similarity metric is based on the Euclidean distance (though other distances could also be used) between the abstracted feature vectors that represent two images, and it is defined as: $d(Q, I) = \sqrt{\sum_{j=1}^{n}(h_j^Q - h_j^I)^2}$ where Q and I represent the query image and one of the images in the image set, and h_j^Q and h_j^I represent the coordinate values of the feature vectors of these images respectively. Note that a smaller distance reflects a closer similarity match. The above inference stems from the fact that color histograms are mapped onto points in a n-dimensional space, and similar images would therefore appear relatively close to each other.

This distribution of colors within an image can be further extended to capture some spatial knowledge about the color distribution of an image. A commonly used extension of the Global Color Histogram approach is its incorporation into a partition based system. The idea of *Partition* or *Grid-based* approaches is to segment the image into cells of equal or varying sizes, depending upon the implementation requirements, while using Local Color Histograms (LCHs) to represent each of these cells. A LCH is computed in the exact same way a GCH is computed, but instead of the whole image, only the cell's contents are used. The Grid approach is therefore an extension of the GCH approach, since each cell is treated like an image by itself and the overall distance between two images becomes a sum of the individual Euclidean distances between corresponding LCHs. Thus, a common similarity metric used in a Grid based approach takes the following form: $d(Q, I) = \sum_{k=1}^{m} \sqrt{\sum_{j=1}^{n}(h_j^Q - h_j^I)^2}$ where all variables remain the same as defined earlier, but k represents the cell number in the grid placed over the image and m is the total number of cells making up the grid itself.

It should be noted though, that in using either the GCH or Grid-based approach, storing n-dimensional vectors of a color histogram for each image in the database may consume significant storage space. In order to minimize the space requirement, we propose the use of a compact representation of these vectors, thereby leading us to the utilization of binary signature bit-strings. An image's signature bit-string, hereafter referred to as *signature*, is an abstract representation of the color distribution of an image by bit-strings of a pre-determined size. The retrieval procedure assumes that the image signatures are stored sequentially in a file. To process a query, the file is scanned and all image signatures are compared against the signature of the query image using a well-defined similarity metric. The candidate images are then retrieved and ranked according to their similarity with the query image.

One should note that, in order to avoid sequential scanning, the signatures need to be indexed if the image database is of a non-trivial size. An ideal index structure would then quickly filter out all irrelevant images. However, the issue of efficiently indexing image signatures, as proposed in this paper, remains for further research, and is not dealt with in this paper. Our main goal is to demonstrate that the proposed signatures are an efficient and compact alternative to using GCHs. Indeed, we will argue later in the paper that the use of the proposed signatures for images may be useful for storing and searching an image database of reasonable size using a relative small amount of main memory. Hence, we might avoid, in some non-trivial cases, the need of a disk-based access structure. Nevertheless, if one aims at indexing several millions of images then a disk-based index is needed. As a matter of fact, we are currently improving a disk-based access structure for binary signatures [TNM00] in order to tackle the problem of indexing a very large number of images efficiently.

The effectiveness of a CBIR system ultimately depends on its ability to accurately identify the relevant images. Our basic motivation is based on the observation that classical techniques based on GCHs often offer poor performance since they treat all colors equally, and take only their distributions into account. More specifically, the relative density of a color is not taken into account by these approaches. Let us motivate this point through a simple example. Consider two different pictures of a single colorful fish, with two quite different and large backgrounds. The similarities/differences among the many less dense colors of the fish are likely more important than the large difference in the background. Hence, we believe that colors relatively less dense should be given more attention. This idea is discussed later in this paper. The first contribution of this paper is the design of two new image abstraction methods using bit-string signatures that exhibit a more efficient image retrieval than GCHs, while still being better in terms of storage space. The best performer is in fact the one methods which gives more "attention" to less dominant colors, confirming our belief above. The second contribution of this paper is to show that the abstraction method can be implemented in a way that very low storage overhead is required. We also show that the use of signatures with a Grid-based approach outperforms the use of LCHs also using the a Grid-based approach.

The rest of the paper is organized as follows. In Section 2, an overview of the current literature in the area is given. Section 3 contains a description of the proposed image abstraction methodologies, as well as the similarity metric devised for the proposed signatures. We also discuss how the approaches fit into Grid-based CBIR. The methodology to evaluate the effectiveness of the retrieval procedure in the proposed and existing techniques, and the obtained experimental results, are given in Section 4. Finally, Section 5 concludes the paper and states the directions for future work.

2 Related Work

In recent years, there has been considerable research published regarding CBIR systems and techniques. In what follows, we give an overview of some representative work related to ours, i.e., color-oriented CBIR.

QBIC [F^+95] is a classical example of a CBIR system. It performs CBIR using several perceptual features e.g., colors and spatial-relationship. The system utilizes a partition-based approach for representing color. The retrieval using color is based on the average Munsell color and the five most dominant colors for each of these partitions, i.e., both the global and local color histograms are analyzed for image retrieval [N^+93]. Since the quadratic measure of distance is computationally intensive, the average Munsell color is used to pre-filter the candidate images. The system also defines a metric for color similarity based on the bins in the color histograms.

A similarity measure based on color moments is proposed in [SO95]. The authors propose a color representation that is characterized by the first three color moments namely color average, variance and skewness, thus yielding low space overhead. Each of these moments is part of an index structure and have the same units, which make them somewhat comparable to each other. The similarity function used for retrieval is based on the weighted sum of the absolute difference between corresponding moments of the query image and the images within the data set. A similar approach was also proposed by Appas et al [A^+99], the main difference being that the image is segmented in five overlapping cells, i.e., much like the grid-based approach discussed earlier. A superimposing 4×4 grid is also used in [SMM99].

Another technique for integrating color information with spatial knowledge in order to obtain an overall impression of the image is discussed in [HCP95]. The technique is based on the following steps: Heuristic rules are used to determine a set of representative colors. These colors are then used to obtain relevant spatial information using the process of maximum entropy discretization. Finally, the above computed information is utilized to retrieve the relevant similar images from the image set. The technique is based on computing the GCH and color histogram for the grid at the center. The histogram considers only those colors that contribute at least a certain percentage of the total region area. The color with the largest number of pixels in the GCH is presumed to be the background color of the image, and the color with the largest number of pixels in the central grid is presumed as being the color of the object in the image. The technique therefore addresses the problem of background distortion of an image object.

A system for color indexing based on automated extraction of local regions is presented in [SC95]. The system first defines a quantized selection of colors to be indexed. Next, a binary color set for a region is constructed based on whether the color is present for a region. In order to be captured in the index by a color set, a region must meet the following two requirements: (i) There must be at least N pixels in a region, where N is a user-defined parameter, and (ii) Each color in the region must contribute with at least a certain percentage of the total region area; this percentage is also user-defined. Each region in the image is represented using a bounding box. The information stored for each region includes the color set, the image identifier, the region location and the size. The image is therefore queried, based not only on the color, but also on the spatial relationship and composition of the color region.

In [SNF00] the authors attempt to capture the spatial arrangement of different colors in the image, based on using a grid of cells over the image and a variable number of histograms, depending on the number of distinct colors present. The paper argues that on an average, an image can be represented by a relatively low number of colors, and therefore some space can be saved when storing the histograms. The similarity function used for retrieval is based on a weighted sum of the distance between the obtained histograms. The experimental results have shown that such a technique yields 55% less space overhead than conventional partition-based approaches, while still being up to 38% more efficient in terms of image retrieval.

Pass et al [PZM96] describe a technique based on incorporating spatial information with the color histogram using coherence vectors (CCV). The technique classifies each pixel in a color bucket as either coherent or incoherent, depending upon whether the pixel is a constituent of a large similarly-colored region. The authors argue that the comparison of coherent and incoherent feature vectors between two images allows for a much finer distinction of similarity than using color histograms. The authors compare their experimental results with various other techniques and show their technique to yield a significant improvement in retrieval. It is important to note that CCVs yield *two* actual color histograms since each color will have a number of coherent and a number of incoherent pixels, therefore CCVs require twice as much space than traditional GCHs, which is a drawback of the approach.

3 Image Abstraction and Retrieval using Signatures

Most of the summary methods described in the previous section improve upon image retrieval by incorporating other perceptual features with the color histogram, spatial information being the most common. A majority of these efforts have been directed toward a representation of only those colors in the color histogram that have a significant pixel dominance. Our best approach differs exactly in this regard. We actually emphasize more on the colors which are less dominant while still taking major colors into account. In order to use signatures for image abstraction, we have designed the following scheme:

- Each image in the database is quantized into a fixed number of n colors, $C = (c_1, c_2, \ldots, c_n)$ to eliminate the effect of small variations within images and also to avoid using a large file due to the high resolution representation [P$^+$99].

- Each color element c_j is then discretized into t binary bins ($B^j = b_1^j\, b_2^j\, \ldots\, b_t^j$) of equal or varying capacities, referred to as the bin-size. If all bins have the same capacity, we say that such an arrangement follows a Constant-Bin Allocation (CBA) approach, otherwise it follows a Variable-Bin Allocation (VBA) approach. As an example, consider an image comprising of n colors and t bins. The signature of this image would then be represented by the following bit-string: $S = b_1^1\, b_2^1 \ldots b_t^1\;\; b_1^2\, b_2^2 \ldots b_t^2\; \ldots\; b_1^n\, b_2^n \ldots b_t^n$ where, b_i^j represents the i^{th} bin relative to the color element c_j. For simplicity, we refer to the sub-string $b_1^j\, b_2^j \ldots b_t^j$ as B^j $(1 \leq j \leq n)$, hence, the signature of an image I can also be denoted as: $S_I = B_I^1\, B_I^2 \ldots B_I^n$.

- The normalized values obtained after automatic color extraction are used within the corresponding set of bins to generate a binary assignment of values indicating the presence or absence of a color with a particular density range. Using the CBA approach, each color c_j has its bins set according to the following condition:

$$b_i^j = \begin{cases} 1 & \text{if } i = \lceil h_j \times t \rceil \\ 0 & \text{otherwise} \end{cases}$$

Note that we assume that the entries in the global color histogram are normalized with respect to the total number of pixels in the image.

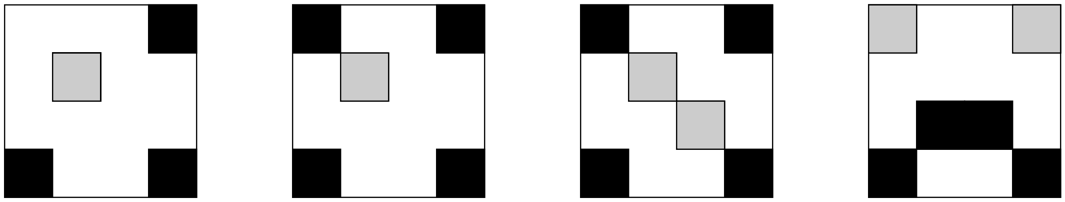


Image A   Image B   Image C   Image D

Figure 1: Sample image set

As an example, consider image A in Figure 1 having three colors, and for simplicity, let us assume n=3, hence, $C = (c_1, c_2, c_3)$ = (black, grey, white). The normalized color densities can then be represented by the vector $H_A = (h_1, h_2, h_3)$ = (0.18, 0.06, 0.76) where each h_j represents the percentage pixel dominance (percentage) of color c_j. Next, assume that the color distribution is discretized into $t = 10$ bins of equal capacities, that is, each bin accommodates one-tenth of the total color representation. Hence, b_1 would accommodate percentage pixel dominance from 1% to 10%, and b_2 accommodates from 11% to 20% and so on. Therefore, image A can then be represented by the following signature (as detailed in Table 1—which also shows the signatures for the other images in Figure 1): $S_A =$ 010000000010000000000000000100.

Color Set of Bins	Color Density	Binary Signature									
		b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
Image A											
c_1/B_A^1	18%	0	1	0	0	0	0	0	0	0	0
c_2/B_A^2	6%	1	0	0	0	0	0	0	0	0	0
c_3/B_A^3	76%	0	0	0	0	0	0	0	1	0	0
Image B											
c_1/B_B^1	24%	0	0	1	0	0	0	0	0	0	0
c_2/B_B^2	6%	1	0	0	0	0	0	0	0	0	0
c_3/B_B^3	70%	0	0	0	0	0	0	1	0	0	0
Image C											
c_1/B_C^1	24%	0	0	1	0	0	0	0	0	0	0
c_2/B_C^2	12%	0	1	0	0	0	0	0	0	0	0
c_3/B_C^3	64%	0	0	0	0	0	0	1	0	0	0
Image D											
c_1/B_D^1	24%	0	0	1	0	0	0	0	0	0	0
c_2/B_D^2	12%	0	1	0	0	0	0	0	0	0	0
c_3/B_D^3	64%	0	0	0	0	0	0	1	0	0	0

Table 1: Detailed Signatures of the images in Figure 1 using CBA

Such an assignment within a set of bins which have equal capacities to accommodate the percentage pixel dominance is referred to as the *Constant-Bin Allocation* (CBA). Another approach, Variable-Bin Allocation (VBA), is based on an assignment where the bins within a set accommodate varying capacities and is presented later in this section.

Recall that each bin is represented by a single bit, therefore the obtained signature is a compact, yet effective, representation of the color content, with due emphasis being placed on the space overhead. To discuss this further, let us assume that the storage of a real number requires f bytes. Therefore, to store a GCH of an image comprising of n colors, $(n \times f)\ bytes$ would be required, whereas the proposed image abstraction requires only $(n \times t)\ bits$ for representing an image. The proposed technique is therefore much more space efficient than GCHs. For instance, by using $n = 64$, $f = 2$ bytes and $t = 10$, the CBA's signature requires 80 bytes per image, whereas the GCH requires 128 bytes. That is to say that CBA's space overhead is 37% smaller than that required by the corresponding GCH. In addition, it is reasonable to expect that a large number of colors may not be present in an image, which may lead to a large sequence of 0's, which would allow the use of compression techniques, e.g., simple run-length encoding [FZ92], to save even more space.

However, a more careful analysis of the signatures reveal that at most one single bit is set within any bin set B^j. Hence, one could simply record the *position*, within B^j, of that one bit which is set instead of recording the whole 10 bit signature for B^j. If each bin set comprises of t bins, only $\lceil \log_2 t \rceil$ bits are needed to encode the position of the set bit (if any). Thus, for $t = 10$, the CBA approach would require a mere 4 bits to encode each color. Revisiting the example above (where $n = 64$, $f = 2$ and $t = 10$), the CBA approach would require only 32 bytes compared to the 128 bytes required by the GCH, a substantial savings of 75% in storage space. If we assume that an image's address in disk consumes 8 bytes, then an image can be abstracted and addressed using 40 bytes. For instance, using 4 Mbytes of main memory one would be able to "index" *and* search a mid-size image database of roughly 100,000 images. Perhaps more importantly, one would be able to do so avoiding the overhead of maintaining a disk-based access structure. This is an important result as it allows easy to implement speedy searches for images in many applications domains, e.g., digital libraries.

However, it is not totally clear whether this encoding approach will always yield better "compression" than, for instance, using run-length encoding on the full signature, as this is highly dependent on the number of consecutive 0's a signature would have. That would depend on the number of colors present in an image – which, as we shall see later, is usually low. Nevertheless, we know that at least the use of $\lceil \log_2 t \rceil$ bits per color is feasible and we shall use this as a lower bound for the savings in the space overhead.

For the sake of clarity, in the remainder of this paper we shall display the full binary signatures for the images involved in the examples. The reader should be aware that such an explicit signature representation is not necessarily translated into the CBIR's storage space.

Once the signatures for each image in the data set is pre-computed and stored into the database, the retrieval procedure can take place. It is basically concerned with computing the associated similarity between the signatures of the user-specified query image and all the other images in the database. At the outset, we used the following measure to analyze the image similarity:

$$d_0(Q, I) \ = \ \sum_{j=1}^{n} \mid (pos(B_Q^j) - pos(B_I^j) \mid$$

where $pos(B_R^k)$ gives the position of the set bit within (the set of bins) B^k of image R. For instance, using image A in Figure 1, we have $pos(B_A^1) = 2, pos(B_A^2) = 1$ and $pos(B_A^3) = 8$. This measure of similarity however is not robust, as illustrated in the following example.

Consider the signatures of images X, Y and Z as detailed in the Table 2. By simple inspection of the color densities (second column in the table) it is clear that images X and Z are more similar to each other than images X and Y. In fact, it seems reasonable to assume that a smaller difference in a few colors is less perceptually critical than a larger difference in a single color – that is exactly what is illustrated in Table 2. However, we have: $d_0(X, Y) \ = \ (4-4) \ + \ (4-4) \ + \ (5-3) \ = \ 2$, and $d_0(X, Z) \ = \ (4-3) \ + \ (4-3) \ + \ (5-5) \ = \ 2$, which suggests that both Y and Z are equally similar to X, thus contradicting our intuition.

Fortunately though, if we square the individual distances between the sets of bins, we can obtain a more robust description of the similarity metric. The new (and hereafter used) distance becomes:

$$d(Q, I) \ = \ \sum_{j=1}^{n} [pos(B_Q^j) - pos(B_I^j)]^2$$

Color/ Set of Bins	Color Density	Binary Signature									
		b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
Image X											
c_1/B_X^1	30%	0	0	0	1	0	0	0	0	0	0
c_2/B_X^2	30%	0	0	0	1	0	0	0	0	0	0
c_3/B_X^3	40%	0	0	0	0	1	0	0	0	0	0
Image Y											
c_1/B_Y^1	39%	0	0	0	1	0	0	0	0	0	0
c_2/B_Y^2	39%	0	0	0	1	0	0	0	0	0	0
c_3/B_Y^3	22%	0	0	1	0	0	0	0	0	0	0
Image Z											
c_1/B_Z^1	29%	0	0	1	0	0	0	0	0	0	0
c_2/B_Z^2	29%	0	0	1	0	0	0	0	0	0	0
c_3/B_Z^3	42%	0	0	0	0	1	0	0	0	0	0

Table 2: Signatures to exemplify the derivation of the similarity metric

The reasoning behind the squaring is to further accentuate the distance between corresponding sets of bins. Using this new definition of distance on the example discussed above, we have: $d(X,Y) = 4$ and $d(X,Z) = 2$, which reflects more closely our assumed perception of image similarity.

Finally, using the obtained similarity distances, the image set is then reordered with respect to their ascending distances (relative to the query image), and the top ranked ones are presented as the query's answer.

For the sake of illustration consider the images of Figure 1, and their respective signatures in Table 1. Using the similarity metric defined above yields: $d(A,B) = 2 < d(A,C) = d(A,D) = 3$, which matches our intuition. However, at this point it is worthwhile to single out the most important disadvantage of using only the color distribution as a basis to abstract an image. It is not unreasonable to argue that images C and D (Figure 1) are not similar. Even though they do have the same color distributions, the colors have very different *spatial* distribution, which is *not* captured when only the quantitative color density is used, in fact, $d(C,D) = 0$.

In such a case, superimposing a grid over the image, obtaining LCHs for each resulting cell and calculating the cumulative distance among such cells (as explained in Section 1) would help to verify that both images are indeed not similar. To illustrate this, Figure 2 shows images C and D "exploded" so that the cells (assuming a 2×2 grid) are explicit (the arrows indicate how cells would be compared to each other). It should be obvious that the sum of the distances between signatures for corresponding cells is not null, therefore those images are color-wise but not spatially similar. It is important to note that the CBA approach and the similarity metric discussed above can be used within each cell as well.

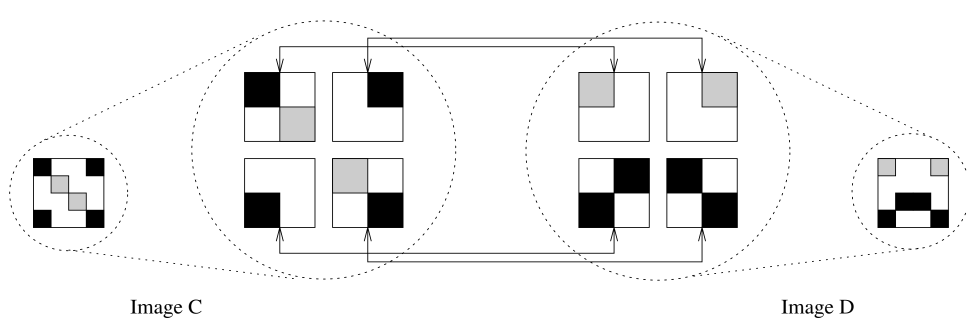


Figure 2: Images C and D from Figure 1 "exploded" in a 2×2 grid

We have observed that less dominant colors form a significant composition in an image. In fact, investigating the 20,000 images used in our experiments, we verified that each image has, on average, 11.4 colors, with a standard deviation of 3.5. In addition, most of such colors (8 on average) have a density of 10% or less. Hence, it is conceivable to conjecture that a substantial cover of an image is due to many colors which individually constitute a small portion of the whole image. This lead us to modify the signatures in order to emphasize colors with smaller densities. Towards that goal, the VBA scheme was developed, and it is discussed next.

Variable-Bin Allocation (VBA) is based on varying the capacities of the bins which represent the color densities. The design of VBA is based on the fact that distances due to less dominant colors are important for an efficient retrieval. The generated signature therefore lays a greater emphasis on the distance between the less dominant colors than on larger dominant colors. The experiments performed on VBA, assuming a color distribution discretized into $t = 10$ bins, were based on b_1, b_2, and b_3, each accommodating 3% of the total color density, and b_4 and b_5 accommodating 5% each. The remaining bins under consideration are based on a reasoning similar to the one used for the the CBA approach, i.e. b_6 accommodates density representations from 20% to 29%, and b_7 accommodates the representations from 30% to 39%, and so on. Note that densities over 60% are all allocated to the same bin (b_{10}). This is due to the fact that an image rarely has more than half of itself covered by a single color. In other words we enhance the granularity of color with smaller distributions, without completely ignoring those with a large presence.

Similar to the previous example where we determined the signatures of images A,B and C using a CBA, the signature of these images using VBA are shown in Table 3. Now, we have $d(A,B) = 1 < d(A,C) = d(A,D) = 5$, which matches our intuition even better, i.e., the use of VBA further increased the distance between image A and both C and D (which are still demeed identical). In addition, note that when using CBA we would have had $d(A,B) = 2 > d(B,C) = 1$ but VBA yields $d(A,B) = 1 < d(B,C) = 4$, reflecting the fact that the same quantitative change (one single cell) within a color which is less dense is more important than the same quantitative change within a more predominant color. This indicates that by using VBA, the distance between dissimilar images is likely to be significantly increased. Our belief that this scheme would enable a more efficient retrieval than CBA was confirmed in the experiments discussed next.

Color Set of Bins	Color Density	Binary Signature b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
Image A											
c_1/B_A^1	18%	0	0	0	0	1	0	0	0	0	0
c_2/B_A^2	6%	0	1	0	0	0	0	0	0	0	0
c_3/B_A^3	76%	0	0	0	0	0	0	0	0	0	1
Image B											
c_1/B_B^1	24%	0	0	0	0	0	1	0	0	0	0
c_2/B_B^2	6%	0	1	0	0	0	0	0	0	0	0
c_3/B_B^3	70%	0	0	0	0	0	0	0	0	0	1
Image C											
c_1/B_C^1	24%	0	0	0	0	0	1	0	0	0	0
c_2/B_C^2	12%	0	0	0	1	0	0	0	0	0	0
c_3/B_C^3	64%	0	0	0	0	0	0	0	0	0	1
Image D											
c_1/B_D^1	24%	0	0	0	0	0	1	0	0	0	0
c_2/B_D^2	12%	0	0	0	1	0	0	0	0	0	0
c_3/B_D^3	64%	0	0	0	0	0	0	0	0	0	1

Table 3: Detailed Signatures of the images in Figure 1 using VBA

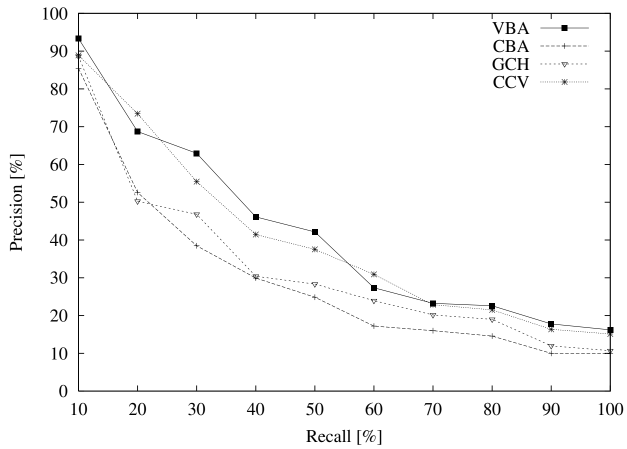


Figure 3: Recall and Precision using the full image

4 Experimental Results

The experiments were performed on a heterogeneous database of 20,000 images from a CD-ROM by Sierra Home (*PrintArtist Platinum*) with 15 images, determined *a priori*, used as queries. The query images where obtained from another collection of images (*Corel Gallery Magic 65,000* CD-ROM) to diminish the probability of biasing the results obtained. Each query image is a constituent of a subset determined (also *a priori*) of similar images[4] The images in each of these subsets resemble each other based on the color distribution and also the image semantics (all subsets can be seen at: `http://www.cs.ualberta.ca/~mn/CBIRdataset/`). The color constituents were normalized to a 64 color representation using the RGB color model. Both CBA and VBA using $t = 10$ were used. Results for the GCH and Color Coherence Vectors (CCV) [PZM96] were also generated to serve as a benchmark. For the sake of illustration, a couple of sample queries and the respective first ten returned images for VBA are shown in the Appendix (results from other other approaches are not shown due to lack of space).

The evaluation of a retrieval system can be performed in various different ways, depending on the ranking of the relevant image set, the effectiveness of indexing, and the operational efficiency etc. The effectiveness of an information retrieval system is often measured using *recall* and *precision* [BYRN99, WMB99] where, recall measures the ability of a system to retrieve relevant documents, and conversely precision is a measure of the ability to reject the irrelevant ones.

Figure 3 shows that the VBA approach delivers the best (higher) precision vs recall curve, specially in the upper-mid range of recall values (60%–80%). The figure also shows that while CCV is more effective than GCH in the first half of recall values, it performs nearly identically in the 2nd half. It is important to note that CCV's space overhead is twice as large as GCH's, given that there are two histograms, one for coherent and another incoherent pixels. Recall that one of our goals in this paper was to reduce the CBIR's space overhead (hence reducing search time) substantially while maintaining the retrieval effectiveness.

Figures 4, 5 and 6 clearly show that the VBA excels the other two approaches for all grid sizes. As we increase the granularity of the grid, the performance of GCHs fairs slightly better than CBA's. CCV was not included in this study because of the larger space overhead and also because it was outperformed by VBA in Figure 3. On the other hand, GCH results were maintained to be used as a yardstick measure.

However, the usefulness of precision and recall measures has been challenged recently. It is a "double"

[4]At this point it is noteworthy pointing out that, to the best of our knowledge, there is no standard benchmark for effectiveness evaluations of different CBIR approaches, thus the *ad-hoc* nature of our evaluation method.

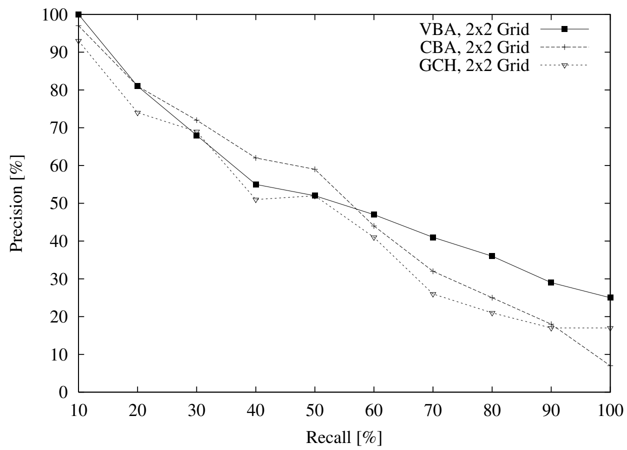


Figure 4: Recall and Precision using a 2×2 grid

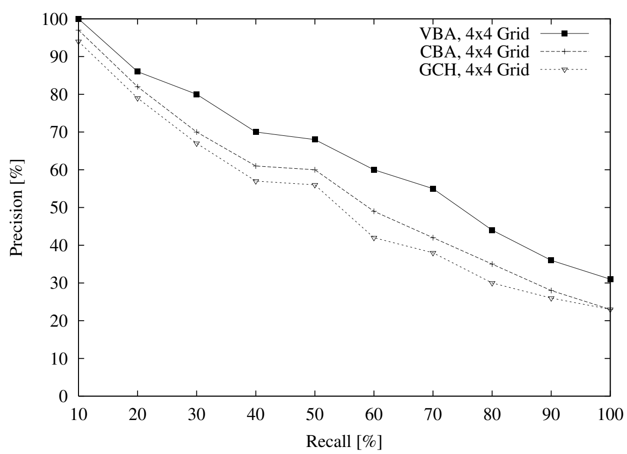


Figure 5: Recall and Precision using a 4×4 grid

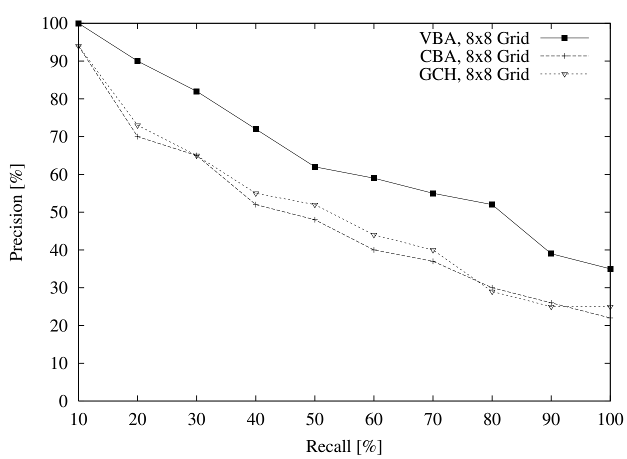


Figure 6: Recall and Precision using a 8×8 grid

measure which makes it harder to compare the results of two different approaches, as, for instance, one particular user may be more interested in precision whereas another may be only concerned with recall. In fact, Su suggests that neither precision nor recall are highly significant in evaluating a retrieval system [Su94].

In order to address this, we have also used a methodology closely related to the one presented in [F+94] to evaluate QBIC's performance. The chosen metric is based on the ratio of the actual average ranking of the similar images to the ideal average ranking of these images in the result set. As an example, consider an image set comprising of ten images including two images that closely resemble a given query image. Now, assume methodology A places the desired images from the subset at positions 3 and 6 thus yielding an average rank of 4.5; and say that the methodology B places them at position 3 and 4, resulting in an average rank of 3.5. Knowing that the ideal average rank would be 1.5 – the desired images would be the first two to appear in the answer set – the retrieval effectiveness of methodology A is 3, and that of methodology B equals 2.33. Thus, methodology B is more effective than methodology A, since the desired results are ranked better in the answer set. The smaller the obtained value, the better the effectiveness. More formally, for a given query image, we have:

$$\text{Retrieval Effectiveness} = \frac{2\ \sum_{i=1}^{s}\ s_i}{(s\ (s+1))}$$

where, s is the number of images deemed, beforehand, similar to the query image, and s_i represents the ranks of such images in the result set. In practice, this ratio attempts to measure how quickly (in relation to the expected answer set size) one is able to find all relevant images. Note that this measure is normalized with respect to the size of the answer sets, thus one can use an average performance as a representative measurement.

Table 4 lists the retrieval effectiveness of the reference images, the average effectiveness criteria, and the respective standard deviation, for analyzing the overall performance of all approaches.

Technique	Retrieval Effectiveness	Success Rate
VBA	24.40	53%
CBA	46.44	07%
GCH	51.87	20%
CCV	30.63	20%

Table 4: Summary of the experimental results

The experimental results show that while both the proposed image abstraction techniques perform better than GCHs, the VBA approach produces significantly better results. Using VBA we obtain approximately 50% (resp. 20%) better retrieval effectiveness than using GCH (resp. CCV), while still saving 75% of storage space (resp. 87.5%). In addition, the table also shows the success rate of the investigate approaches, i.e., the percentage of queries for which each approach delivered the best performance. VBA is the clear winner. Given our stress in saving storage space it is important to also consider that CCV is the most expensive of the four approaches benchmarked. Hence we do not use CCVs any further with the Grid based approach.

Technique	Retrieval Effectiveness	Success Rate
VBA	8.20	67%
CBA	11.47	13%
GCH	20.04	20%

Table 5: Summary of the experimental results using a 2×2 grid

Tables 5, 6 and 7 show the retrieval effectiveness values obtained when using the grid-based approach. Overall, the results are better than those obtained without utilizing a grid-based approach (i.e., whole image abstractions). Again, the VBA approach is clearly the best performer, being about 50% more efficient than the other

Technique	Retrieval Effectiveness	Success Rate
VBA	4.00	53%
CBA	8.56	20%
GCH	8.72	27%

Table 6: Summary of the experimental results using a 4×4 grid

approaches. On the other hand, CBA's earlier advantage over GCHs is lost in terms of success rate, and is slightly worse in terms of retrieval effectiveness in the case of the 8×8 grid. There is an interesting result though, which is not clearly seen from the precision vs recall curves shown earlier. Making the grid finer does not improve performance linearly. In fact, when doubling (halving) the grid (cell) size from 4×4 to 8×8, the relative performance nearly did not change (even though the VBA's success rate did improve). As a finer grid implies in larger computational effort, e.g., more signatures to compute *and* store, more comparisons among signatures, etc, we are inclined to say that a medium resolution grid (e.g., 4×4) is a good comprise between a system's resources and performance.

Technique	Retrieval Effectiveness	Success Rate
VBA	4.44	73%
CBA	9.01	7%
GCH	7.74	20%

Table 7: Summary of the experimental results using a 8×8 grid

It is important to stress however, that the use of such a grid approach has a serious drawback. The image representation is not invariant to rotation and/or translation any longer. This is important as an image and its upside-down copy might be deemed quite different, whereas most users would judge them identical, but with a different orientation. The bottomline is that the use of a grid-based approach to enhance retrieval effectiveness and search precision is not as robust as when not using it. Nevertheless, should the user be willing to lose rotation/translation invariance, such an approach delivers very good results as discussed above.

5 Conclusions

The main contribution of this paper is the design of two variations of a new image abstraction methodology to generate a signature for an image content. We have also introduced a metric for ranking the results, based on the comparison of signatures between a query image and the image collection. In general, the two image abstraction techniques are based on a compact representation of the information in a global color histogram by decomposing the color distributions of an image into bins that accommodate varying percentage compositions. The use of these proposed methodologies also result in a space reduction due to the bitwise representation of the image content.

The experiments performed on a heterogeneous database of 20,000 real images with 15 images used as queries, demonstrate that while both of the image abstraction techniques perform better than using Global Color Histograms, and the retrieval process using a Variable-Bin Allocation (VBA) produces the best results – approximately 50% better than the Global Color Histogram and up to 20% better than Color Coherence Vectors. In addition, using the proposed binary signatures, one can save at least 75% of storage space when compared to storing Global Color Histograms and a substantial 87.5% when storing Color Coherence Vectors. This result is important in the sense that it allows one to have good retrieval effectiveness as well as avoid the use (and implementation) of complex disk-based access structures. Thus we envision one can easily improve existing image processing/management applications with our approach, allowing users to search through a reasonable number of images using not much of his/her computer resources.

We have also observed that superimposing a grid on the image may indeed improve retrieval effectiveness for all approaches, in which case a medium sized grid is probably the best comprise. Again, using such a grid approach has the drawback that the CBIR system would be sensitive to image rotation for instance, which may detract from the overall quality of the system, but if the user is aware of this limitation, it can indeed produce good results.

Among the possible venues for future research, we should focus on the following ones: using the CBA/VBA approach with a smaller number of bins, e.g., collapsing the last bins (as we have argued that very few colors show a large distribution), and, perhaps more important, designing a hash or tree-based access structure to support the signature proposed in this paper, and speedup query processing. Another interesting opportunity for research would be devising ways to make the grid-based approach less invariant to rotations. Finally, a more comprehensive set of tests, including the signature encoding/compression issue and comparisons to other proposed CBIR techniques, are planned for future research.

Acknowledgements

The authors would like to thank R. O. Stehling for organizing the dataset which was used for evaluating the proposed retrieval methods and for discussions about signature compression. V. Oria's feedback was also appreciated. M. A. Nascimento was partially supported by a Research Grant from NSERC Canada. C. Mastaller's work was supported by a 2000 NSERC Canada Summer Research Award.

References

[A+99] A.R. Appas et al. Image indexing using composite regional color channels features. In *Proc. of SPIE – Storage and Retrieval for Image and Video Databases VII*, volume 3656, pages 492–500, 1999.

[BYRN99] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[dB99] A. del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann, 1999.

[F+94] C. Faloutsos et al. Efficient and effective querying by image content. *J. of Intelligent Information Systems*, 3(3/4):231–262, 1994.

[F+95] M. Flickner et al. Query by Image and Video Content: The QBIC System. *IEEE Computer*, pages 23–32, 1995.

[FZ92] M.K. Folk and B. Zoellick. *Files Structures*. Addison-Wesley, 2nd edition, 1992.

[HCP95] W. Hsu, T.S. Chua, and H.K. Pung. An Integrated Color-Spatial Approach to Content-Based Image Retrieval. In *Proc. of ACM Multimedia 1995*, pages 305–313, San Francisco, USA, 1995.

[N+93] W. Niblack et al. The QBIC Project: Querying Images by Content using Color, Texture and Shape. In *Proceedings of the SPIE - Storage and Retrieval for Image and Video Databases*, pages 173–187, San Jose, USA, 1993.

[P+99] D.-S. Park et al. Image Indexing using Weighted Color Histogram. In *Proc. of the 10th Intl Conf. on Image Analysis and Processing*, Venice, Italy, 1999.

[PZM96] G. Pass, R. Zabih, and J. Miller. Comparing Images Using Color Coherence Vectors. In *Proc. of ACM Multimedia Conf.*, pages 65–73, Boston, USA, 1996.

[SB91] M.J. Swain and D.H. Ballard. Color Indexing. In *Intl J. on Computer Vision*, pages 11–32, 1991.

[SC95] J.R. Smith and S.-F. Chang. Tools and Techniques for Color Image Retrieval. In *Proc. of the SPIE Storage and Retrieval for Image and Video database IV*, pages 40–50, San Jose, USA, 1995.

[SMM99] E. Di Sciascio, G. Mingolla, and M. Mongiello. Content-based image retrieval over the web using query by sketch and relevance feedback. In *Proc. of Proc. of* 4^{th} *Intl. Conf. on Visual Information Systems*, pages 123–130, 1999.

[SNF00] R.O. Stehling, M.A. Nascimento, and A.X. Falcão. On 'shapes' of colors for content-based image retrieval. In *Proc. of the Intl. Workshop on Multimedia Information Retrieval*, pages 171–174, 2000.

[SO95] M. Stricker and M. Orengo. Similarity of Color Images. In *Proc. of the SPIE - Storage and Retrieval for Image and Video Databases III*, pages 40–50, San Diego/La Jolla, USA, 1995.

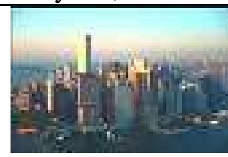
[Su94] L.T. Su. The relevance of recall and precision in user evaluation. In *J. of the American Soc. for Information Science*, pages 207–217, New York, USA, 1994.

[TNM00] E. Tousidou, A. Nanopoulos, and Y. Manolopoulos. Improved methods for signature-tree construction. *Computer Journal*, 43(4):301–314, 2000.

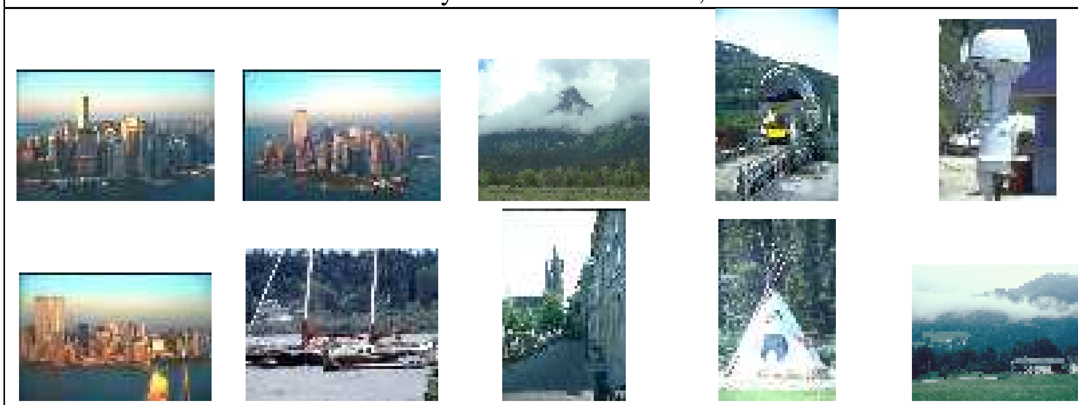
[WMB99] I.H. Witten, A. Moffat, and T.C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, 1999.

Appendix – Sample query/answer sets

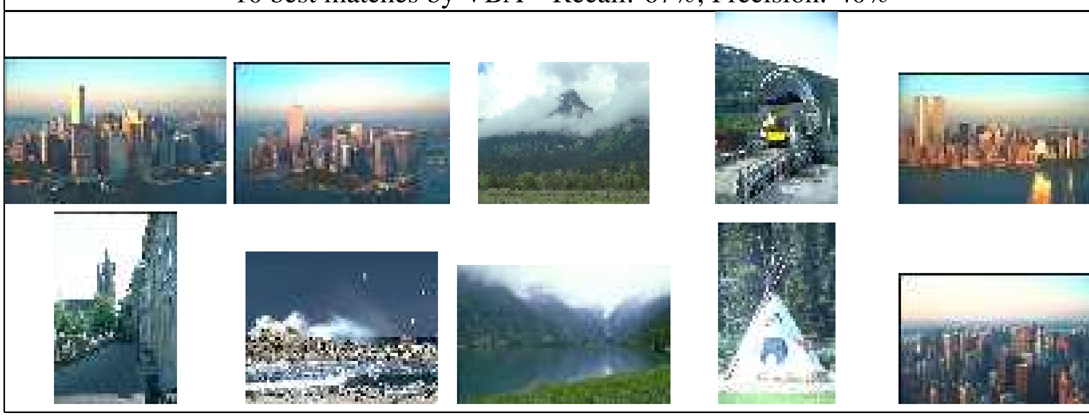
Query Image (Manhattan's skyline, size of expected answer: 6 images)



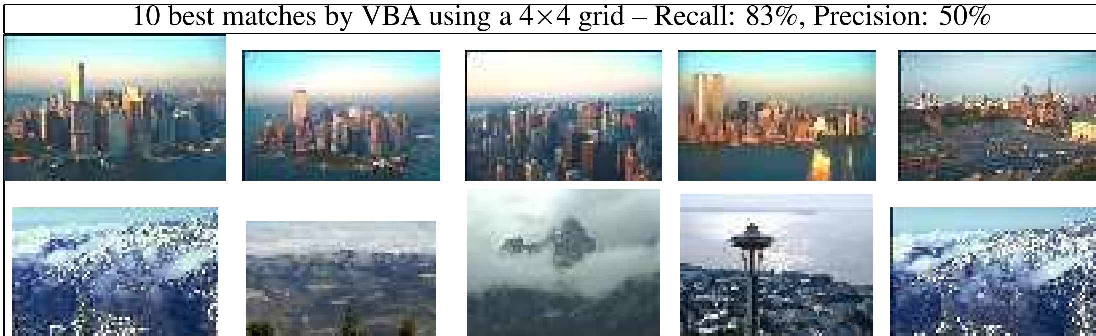
10 best matches by GCH – Recall: 50%, Precision: 30%



10 best matches by VBA – Recall: 67%, Precision: 40%

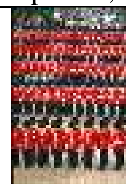


10 best matches by VBA using a 4×4 grid – Recall: 83%, Precision: 50%



Appendix (cont.)

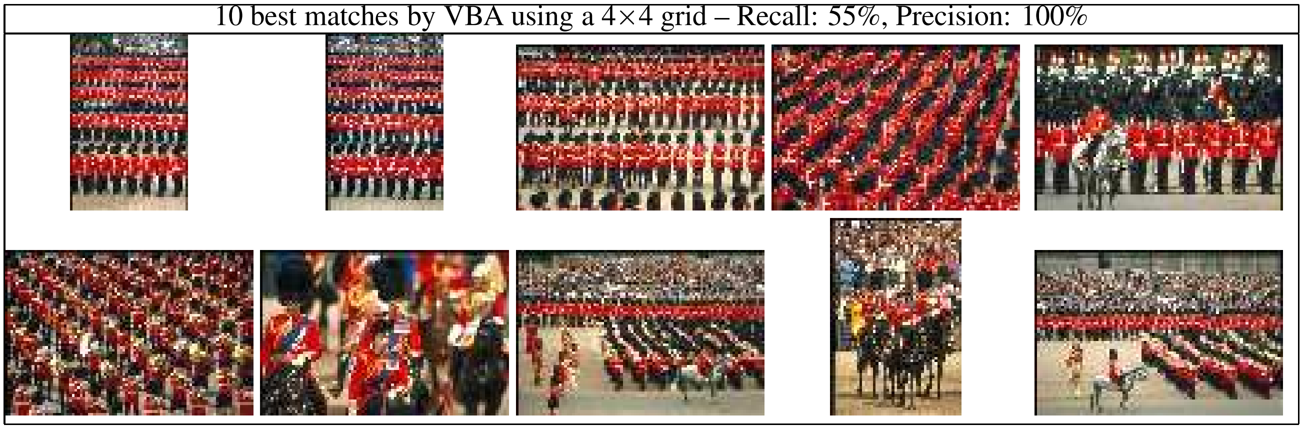
Query Image (Queen's guard parade, size of expected answer: 18)



10 best matches by GCH – Recall: 33%, Precision: 50%

10 best matches by VBA – Recall: 44%, Precision: 80%

10 best matches by VBA using a 4×4 grid – Recall: 55%, Precision: 100%



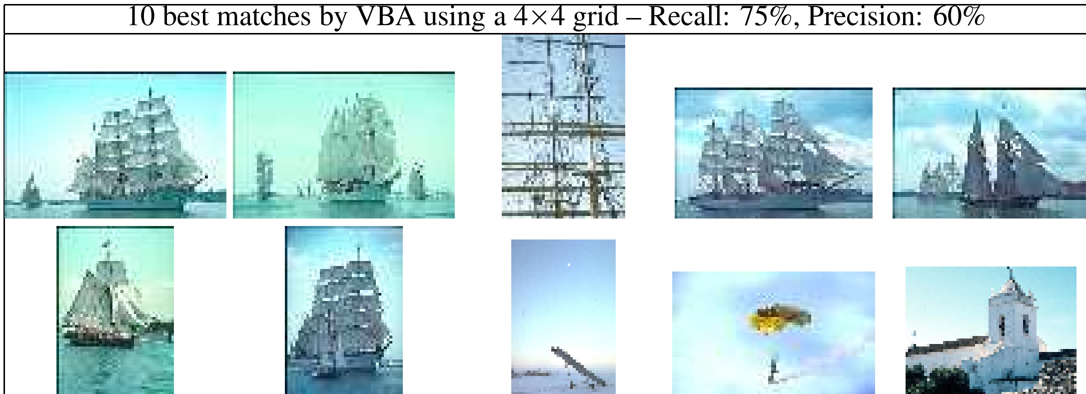
Appendix (cont.)

Query Image (sailing boat, size of expected answer: 8)

10 best matches by GCH – Recall: 50%, Precision: 40%

10 best matches by VBA – Recall: 87%, Precision: 80%

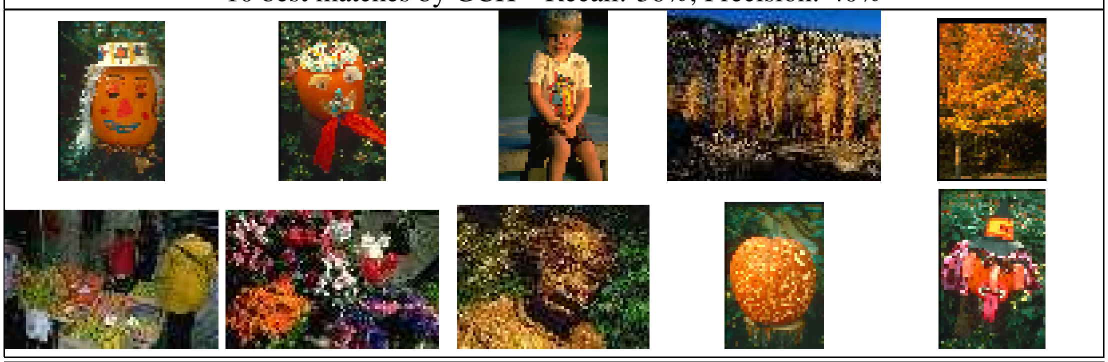
10 best matches by VBA using a 4×4 grid – Recall: 75%, Precision: 60%



Appendix (cont.)

Query Image (Halloween Pumpkins, size of expected answer: 11)

10 best matches by GCH – Recall: 36%, Precision: 40%



10 best matches by VBA – Recall: 27%, Precision: 30%

10 best matches by VBA 4×4 – Recall: 55%, Precision: 60%