# Evaluating Retrieval Effectiveness of Descriptors for Searching in Large Image Databases

Petrina A. S. Kimura[1], João M. B. Cavalcanti[1], Patricia C. Saraiva[1],
Ricardo da S. Torres[2], Marcos A. Gonçalves[3]

[1] Universidade Federal do Amazonas, Brazil
{petrina, john, patricia}@dcc.ufam.edu.br
[2] Instituto de Computação, Universidade Estadual de Campinas, Brazil
{rtorres}@ic.unicamp.br
[3] Universidade Federal de Minas Gerais, Brazil
{mgoncalv}@dcc.ufmg.br

**Abstract.** This article presents an evaluation of image descriptors for searching in large image databases. Several image descriptors proposed in the literature achieve high precision levels when experimented in small (less than 20,000 images) and well-behaved image databases. Our assumption is that retrieval effectiveness may be strongly affected by variations in size, quality, and diversity of the images in the database. In order to verify whether an image descriptor maintains its retrieval effectiveness in large databases, experiments were carried out using several image descriptors and three image collections, including one with over 100,000 images collected from the Web. The results obtained show that in general the retrieval effectiveness of the different descriptors varies little in small image collections whereas in large image collections they differ significantly. Among the descriptors used in the experiments, there are two proposed by us for being used in large and heterogeneous image databases. The proposed descriptors outperform significantly the other descriptors used as baselines in the Web collection. These results give us a better understanding about the features and the strategies that should be followed to construct descriptors for practical search tasks in large image databases.

Categories and Subject Descriptors: H.2.8 [**Database applications**]: Image databases

Keywords: content-based image retrieval, experimental evaluation, image databases

## 1. INTRODUCTION

Advances in multimedia technologies have increased the availability of digital images. This huge amount of visual information is driving the need for more effective and efficient methods to store and retrieve multimedia content. Most earlier research work with Content-Based Image Retrieval (CBIR) focused on developing global approaches for extracting image features [Stehling et al. 2002].

One particular reason was the computational power required by partition-based approaches, in which image features are obtained by computing attributes from several partitions of an image. With recent advances in processing power and more availability and lower cost of storage space (both main memory and disk space), approaches that exploit locality of information such as the partition-based approaches became a viable alternative to global approaches aiming to improve the effectiveness of search results.

Several image descriptors proposed in the literature achieve high levels of efficiency and accuracy. However most of them run experiments using small image databases (less than 20,000 images) [Manjunath et al. 2001] [Zagoris et al. 2010]. Most of these image databases are composed of well defined

The first author is funded by the Amazonas State Research Council(FAPEAM). We also acknowledge the financial support given by CNPq, CAPES, FAPESP and FAPEMIG.

categories, which facilitate the search task resulting in high precision levels. According to the experiments in [Penatti and da S. Torres 2008] the overall effectiveness of image descriptors is relatively low in large and heterogeneous databases with no knowledge or previous categorization of the images. Recently some authors started to study the problem of CBIR on large image databases [Hörster et al. 2007] [Douze et al. 2009]. However, such work is mainly focused on efficiency aspects, such as indexing time, memory space and query time. The vast majority do not evaluate the retrieval effectiveness of their methods using real users on the large image collections.

In this article we report experimental results on the precision levels achieved by several descriptors when used for searching on large and heterogeneous image databases. The experiments were conducted using two small categorized image datasets and a large and heterogeneous image database with over 100,000 images collected from the Web. The results obtained show that in general the retrieval effectiveness of the different descriptors varies little in small image collections whereas in large image collections they differ significantly. Among the descriptors used in the experiments, there are two proposed by us aiming at large and heterogeneous image databases. The proposed descriptors outperform significantly the other descriptors used as baselines. This gives us a better understanding about the features and the strategy that should be followed to construct descriptors for practical image search tasks.

In more details, the proposed image descriptor - Local Color Pixel Classification (LCPC) - is a partition-based approach for image retrieval that uses a simple partitioning scheme, but very effective for searching in large and generic image databases. Our approach incorporates color and spatial information dividing the image into five non-overlapping partitions: one central and the four corners. Our main goal is to provide a simple way for separating foreground objects from the image background based on the assumption that the object of interest is, in general, in the center of an image. After dividing the image, feature vectors are extracted for each partition individually. The similarity between two images considers the similarity between each corresponding partition. As we assume that the center of the image usually defines its semantic, a weighting scheme for the partitions is provided to give to the central partition a higher impact in the final similarity. In addition we also propose an extension of LCPC to include shape features. The motivation for combining color and shape features is to provide more diversity in the answers, for example finding relevant images of different colors of the image query.

In sum, the main contributions of this work are twofold: (i) the evaluation of several descriptors in large and heterogeneous collections with the goal of demonstrating that in such environment the differences among these descriptors will be more prominent than in the very well behaved collections in which they are usually tested; and (ii) the proposal and evaluation of very simple, yet very effective, descriptors for such "hostile" environments along with a discussion of implications of these results.

This article is organized as follows. Section 2 presents a brief overview of content-based image retrieval, including the main concepts useful to this work. Section 3 discusses the descriptors used in the experiments and Section 4 discusses the methodology used for our experiments and achieved results. Finally, section 5 presents final remarks and future work directions.

## 2. OVERVIEW OF CONTENT-BASED IMAGE RETRIEVAL

In recent years, much work has been done in developing the Content-Based Image Retrieval (CBIR) field to support the development of image search engines. However, given the complexity of image analysis, in particular feature extraction and representation, many existing techniques focused on extracting the visual characteristics based on color features.

Color is the most dominant and distinguishing low-level feature used in current CBIR systems. This is not surprising since color is the most straightforward feature used by humans for visual recognition and discrimination. Color features can be represented by a color histogram [Swain and

Ballard 1991], or other proposed color matching technique such as color moments [Stricker and Orengo 1995]. Although these techniques provide satisfactory forms of characterizing color information, they suffer from a complete lack of spatial information of the visual content of an image. As a result they fail to distinguish images with the same color but with a different color distribution.

One way of including some form of spatial information about the visual content of images is to use local color descriptors. The existing local color descriptors can be classified into two main groups: (1) regional approaches and (2) partition-based approaches. Regional approaches divide the image into regions, which may be different for each image. Although these approaches have been shown to be effective [Pass et al. 1996], they often use image segmentation algorithms that have, in general, high computational costs and complex extraction processes of visual features, which make their application in large image search engines unfeasible due to performance issues. Even though the image features of the whole image database are extracted off-line, this technique is not suitable for real and popular image search engines, when several image queries must be segmented during query time and answered in the shortest time possible.

Partition-based approaches, on the other hand, include spatial information of visual features by partitioning the image into fixed-size blocks and then extracting the features of each block individually. This technique is simpler than region-based approaches, since the same partitioning scheme is applied to all images. In the case of large and heterogeneous collections where it is not possible to assume any *a priori* knowledge about the images being analyzed, partition-based approaches seem to be a good choice due to their simple mechanism, taking advantage of spatial information about the visual features of images.

The main contribution of this article is an evaluation of several descriptors, commonly cited in literature, in order to compare their behavior in small and large image databases. Additionally, we propose a new descriptor which is a simple, but very effective partition-based technique, called Local Color Pixel Classification (LCPC), that is suitable for large and generic image collections. Our motivation is based on the assumption that image descriptors presented in the literature work well for small databases with carefully selected images. But problems arise when they are applied on large and heterogeneous databases with no categorization or any previous knowledge about the content of the images. The main goal of our work is to show that visual descriptors previously proposed have good retrieval effectiveness when tested on small and well-behaved image collections, but when applied on large and heterogeneous databases they have their retrieval effectiveness considerably diminished.

## 3. THE IMAGE DESCRIPTORS SELECTED FOR EVALUATION

In this section we describe the image descriptors we have evaluated in this article. These descriptors extract color properties and most of them are partition-based. First, we describe three descriptors belonging to the MPEG-7 standard: Scalable Color Descriptor (SCD), Color Layout Descriptor(CLD), and Edge Histogram Descriptor (EHD) given that they are widely used in the literature [Sikora 2001] [Wong et al. 2007] [Cieplinski 2001]. We evaluated the Color and Edge Directivity Descriptor (CEDD) and Fuzzy Color and Texture Histogram (FCTH) because they outperform the MPEG-7 descriptors [Zagoris et al. 2010]. The Local Color Histogram (LCH) is included as it is the original partition-based image descriptor. We also include the descriptor Border/Interior Pixel Classification (BIC) in our experiments because it presents better results in terms of efficiency and effectiveness for large and heterogeneous image collections when compared with other color descriptors commonly described in the literature [Penatti and da S. Torres 2008]. In Section 3.8, our approach, called Local Color Pixel Classification (LCPC), which is suitable for large image databases is described in detail. A second descriptor that combines LCPC with shape properties (LCPC+Edge) is also described. They were included in the evaluation and compared to the other descriptors listed here.

### 3.1 Scalable Color Descriptor - SCD

The Scalable Color Descriptor [Manjunath et al. 2001] is a Color Histogram which is encoded by a Haar transform with a uniform quantization of the HSV space to 256 bins. The bin values are non-uniformly quantized to a 11-bit value. Its binary representation is scalable in terms of bin numbers and bit representation accuracy over a broad range of data rates. The histogram values are truncated into a 11-bit integer representation and then mapped into a non-linear 4-bit representation in order to achieve a more efficient encoding and to give a higher significance to the small values with high probability. This descriptor uses *L1* distance metric to calculate the similarity between two images.

### 3.2 Color Layout Descriptor - CLD

The Color Layout Descriptor [Manjunath et al. 2001] was developed to capture the spatial distribution of color of any region in the image. As the first step, the image is divided into a grid of 64 blocks, then the dominant color of each block is extracted, usually, through the medium of color. After that, each component of the color space YCrCb is transformed using a cosine transform (DCT), thus generating three sets of coefficients. Finally, a weight is assigned to each coefficient, thus producing a feature vector with information about the predominant colors in each block. To calculate the similarity between two images A and B, two sets of coefficients are considered where the first set belongs to the image A and the second one belongs to image B: A = {DY, DCb, DCr} and B = {DY', DCb', DCr'}. The similarity between A and B is equal to $D(A,B) = \sqrt{\sum_i w_{yi}(DY - DY')^2} + \sqrt{\sum_i w_{bi}(DCb_i - DCb'_i)^2} + \sqrt{\sum_i w_{ri}(DCr_i - DCr'_i)^2}$, where ($\mathrm{DY}_i$, $\mathrm{DCb}_i$, $\mathrm{DCr}_i$) represents the $i^{th}$ DCT coefficient of the respective components of each color. The weights are properly assigned, and a component with lower frequency, receive greater weight.

### 3.3 Edge Histogram Descriptor - EHD

The Edge Histogram Descriptor [Manjunath et al. 2001] divides the image in blocks of 4×4 sub-images to capture five types of edges of each block. Edges are broadly grouped into five categories: vertical, horizontal, 45 diagonal, 135 diagonal, and non orientation specific. Thus each local histogram has five bins corresponding to the above five categories. The EHD uses five filters to compute five edge strengths, if the maximum of these edge strengths exceeds a certain threshold, then the corresponding image block is considered an edge block. The *L1* distance is used to compute the similarity between two edge histogram.

### 3.4 Local Color Histogram - LCH

The Local Color Histogram (LCH) [Swain and Ballard 1991] is a partition-based descriptor which divides the image into fixed blocks of size 4×4. For each block, a color histogram is computed in a quantized 64 color space. The distance function compares the histograms of the corresponding blocks of two images. In our experiments, LCH generated feature vectors with 1024 bins and the Histogram Intersection distance function was used.

### 3.5 Color and Edge Directivity Descriptor - CEDD

The Color Edge Directivity Descriptor is a composite image descriptor [Chatzichristofis and Boutalis 2008] that captures and relates shape, texture, and color from an image. In this descriptor we can consider the full image or an image block. The texture block receives the input block in the YIQ color space and applies the EHD descriptor to construct a histogram of 6 bins, five corresponding to the types of edges found in the image plus one for no edges of any type found. Then, given a threshold, an edge may fall in more than one of the five directional bins. This determines its texture. If the edge does not fall in any edge category then it belongs to the last bin, corresponding to no edges.

For color, each input block is processed in the HSV color space according to the types of edges found previously. The first step is to map each edge block in a preset 10 color bins histogram (Black, Gray, White, Red, Orange, Yellow, Green, Cyan, Blue, and Magenta) using a Binary Haar Wavelet descriptor and a 20 fuzzy-linking rules method. In a second stage this histogram is expanded into a 24 bin color histogram by using Coordinate Logic Filters (CLF) for vertical edge detection in all three HSV channels: Hue is divided into 8 areas: Red to Orange, Orange, Yellow, Green, Cyan, Blue, Magenta, and Blue to Red; Saturation is divided into two fuzzy regions defining the shade of a color based in white; Value channel is divided into three areas: one defines when the pixel (block) will be black and the other two, in combination with Saturation, when it will be gray. Based on this area division a set of 4 fuzzy-like rules are applied transforming the previous 10 color into a 24 color bin histogram comprehending Black, Gray, White, Dark Red, Red, Light Red, Dark Orange, Orange, Light Orange, Dark Yellow, Yellow, Light Yellow, Dark Green, Green, Light Green, Dark Cyan, Cyan, Light Cyan, Dark Blue, Blue, Light Blue, Dark Magenta, Magenta, and Light Magenta. Color information processed to every edge type yields a 6×24=144 bins histogram. The similarity between two images is measured using the Tanimoto Coefficient.

### 3.6 Fuzzy Color and Texture Histogram - FCTH

The Fuzzy Color and Texture Histogram [Zagoris et al. 2010] is also a composite descriptor that resembles the CEDD aiming to capture the image texture, shape, and color. But unlike CEDD, the FCTH capture the texture information through the Haar Transform. The remaining procedure for the color unit is similar to the CEDD descriptor and takes into consideration each coefficient computed in the texture unit. Therefore, the FCTH results in a 8×24 = 192 bin histogram. The similarity measure used by FCTH is the Tanimoto Coefficient.

### 3.7 Border-Interior Pixel Classification - BIC

BIC is a global image descriptor which has presented significant results with respect to efficiency and effectiveness [Stehling et al. 2002]. BIC is suited to diverse and large image databases. It uses RGB color-space quantized in 4×4×4=64 colors and it classifies pixels as either border or interior. A pixel is considered as border if at least one of the four neighboring pixels (top, bottom, left, and right) has a different quantized color. A pixel is classified as interior when all four neighbors have the same quantized color. Two global histograms are created: one for border pixels and another for interior pixels. This technique gives an idea of how the pixels are distributed over the image capturing a notion of texture from the whole image. For example, two images may present similar colors but one contains several small objects resulting in many border pixels. If the other image contains one large object with the same color it would have mainly interior pixels. Therefore the similarity of these two images would not be very high, even though they present similar colors.

BIC also has a fast distance function, namely *dLog* [Stehling et al. 2002] which allows to compare histograms. The *dLog* function calculates the difference between the logarithm of the elements present in a histogram in order to reduce the negative effect introduced when a few elements have very large values and others have low values. These high values would dominate the difference between two histograms. Since these pixels are usually part of the background which covers the majority of the image area, in general they do not determine the semantic of the image. The *dLog* function is used to minimize this distortion. It is defined as:

$$dLog(a,b) = \sum_{i=0}^{i<M} |f(a[i]) - f(b[i])| \qquad (1)$$

$$f(x) = \begin{cases} 0, & \text{if } x = 0 \\ 1, & \text{if } 0 < x \leq 1 \\ \lceil log_2 x + 1 \rceil, & \text{otherwise} \end{cases} \qquad (2)$$

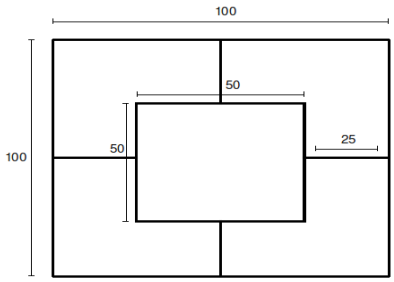



Fig. 1. LCPC partitions size.

In the Equation 1, $a$ and $b$ are two histograms with $M$ bins. Each bin represents a quantized color. The value a[i] represents the $i^{th}$ bin of histogram $a$ and b[i] represents the $i^{th}$ bin of histogram $b$.

When comparing BIC with other color descriptors commonly described in the literature, it presented better results [Penatti and da S. Torres 2008], in terms of efficiency and effectiveness. The use of BIC enables a fast feature extraction and a compact representation of the visual features, not requiring a large amount of storage space. However, as a global descriptor, the feature vector does not represent the concentration of pixels in some regions of an image. Based on this observation our proposal is to define a partition-based descriptor that exploits this locality information of the pixel distribution. Our assumption is that it will present even better results compared to the original BIC.

### 3.8 Local Color Pixel Classification - LCPC

The descriptor LCPC divides the image into five non-overlapping partitions as shown in Figure 2. The central partition corresponds to 50% of total image size and tends to capture the information presented at the center of the image. The background is divided into four equal parts. Figure 1 shows the size of the partitions a given image size of 100×100 pixels.

The four partitions of the background represent the regions corresponding to the top left, top right, bottom left, and bottom right. The idea behind this proposal is that, in general, images often have the object of interest centered in the image, representing the foreground, and other objects are present in the background. Different partitioning schemes were tested in our work (including 2 and 3 partitions) in order to discover which approach presents better effectiveness among other partition arrangements. The five partition scheme obtained better results than the other tested schemes.

If we consider that images can be separated in terms of background and foreground, it is possible to know which colors are predominant in each region. In our approach, we suppose that the foreground information of an image is present in its central partition and the background is present in the remaining partitions. As we assume that the foreground better defines the semantic of an image, our intuition is that the central partition should be the most effective to determine the similarity among

Fig. 2. The LCPC five partitions.

images than the other partitions. Thus, we assign to this partition a higher weight.

In our preliminary experiments, the best weight distribution was 0.7 to the central partition and 0.3 equally divided among the remaining partitions. When computing the similarity between two images, we perform a linear combination for each partition. This ensures that the central partition has a higher impact on the final similarity results.

3.8.1 *Image Representation and Similarity Distance.* Before the extraction process starts, the image is uniformly quantized in the RGB color-space to reduce the number of distinct colors to 64. This scheme is widely adopted in practice and seems to be an appropriate way to reduce the size of the feature vector [Stehling et al. 2002].

Once colors are quantized, the image is partitioned as described above and the color features of each partition are extracted. The extraction process is the same used by the BIC descriptor. Pixels belonging to each partition are classified, according to their neighborhood, as border or interior. A pixel is classified as border if at least one of the 4-neighbors has a different quantized color or it belongs to the border of partition. If its 4-neighbors have the same quantized color it is classified as interior, as detailed in Section 3.7.

After the pixels are classified, color histograms for each partition are computed regarding the pixel classification. In this case, the image is represented in terms of border and interior pixels for each quantized color in each partition.

The similarity distance between two images is given by the *dLog* function proposed in [Stehling et al. 2002]. We use this metric because it has been proved to be fast and more effective than other distance metrics commonly presented in the literature, such as *L1*, *L2*, or Hamming distance [Stehling et al. 2002] [Penatti and da S. Torres 2008]. The *dLog* function compares histograms in a logarithmic scale and its use allows a compact representation for the histograms, in which the histograms bins are normalized in the range $[0, 9]$ as in [Stehling et al. 2002] for compact feature representation. Thus, each histogram bin can assume only 10 distinct values and these values can be stored using 4 bits only ($10 < 2^4$).

This log-based representation allows a reduction of 50% in the required storage space for any histogram-based CBIR approach, requiring only 4 bits of storage per histogram bin. In the case of the LCPC approach, two histograms are generated for each one of the five partitions. Each histogram for each partition has 128 bins (64 for border pixels and 64 for interior pixels). Thus, the LCPC feature vector has a total of 640 bins, which can be stored in 320 bytes.

3.8.2 *Adding Shape Features to the LCPC descriptor.* Color-based descriptors tend to retrieve non-relevant images when the colors of the foreground and background are similar, regardless of other features. Aiming at retrieving other relevant images with different colors and at the same time reducing the number of non-relevant images, we propose a combination of the color features of the LCPC descriptor with shape features to enrich the search capabilities of our descriptor. For this purpose we adapted the Canny edge detection algorithm [Canny 1986], which has become one of the standard edge detection methods [Mai et al. 2008] [Azernikov 2008].

The idea is to detect the image edges and their directions. Given the large amount of possible edge directions, we turn them into a discrete set of four directions: $0, 45, 90$ and 135 degrees. Hence a four-bin edge histogram represents the amount of edges in each direction. The distance function used for evaluating similarity is the Chi-square. Contrary to the idea of the *dLog* function to reduce large differences from distinct histograms, here any large difference in the amount of edges is important and must be maintained. The similarity among two histograms $h$ and $g$ is given by:

$$D(h,g) = \sum_{i=0}^{i<N} \frac{(h[i]-g[i])^2}{h[i]+g[i]} \tag{3}$$

where $N$ is the size of the histogram.

In order to consider both the color and shape features, we combine them using a weighted average of the color and shape histograms distances previously calculated. A weight $t$ is applied to the normalized (values between 0 and 1) Chi-square distance $DQ$. The LCPC weight is equal to $1-t$ and is applied to the normalized (values between 0 and 1) *dLog* distance $DL$. Thus we have the final distance calculated as $D = t \times DQ + (1-t) \times DL$. In Section 4 we present a discussion on how to define the value for $t$ used in our experiments.

## 4. EXPERIMENTS

### 4.1 Experimental Setup

In order to evaluate the retrieval effectiveness of the descriptors presented in Section 3, experiments were carried out using three image databases. The first database is the MPEG-7 CCD (Common Color Dataset) [Zagoris et al. 2010] which consists of $5,466$ color images and a set of 50 queries with their respective ground truth images. The second one is the Wang database [Li and Wang 2003] which is a subset of $1,000$ manually-selected images from the Corel stock photo database having 10 pre-defined classes of 100 images each. In the experiments, one image from each class was used as image query. The third dataset (denoted here as $100K$) consists of $103,348$ images collected from the Yahoo! directory. An enlarged version of our image database has already been used in other research work such as [Penatti and da S. Torres 2008]. This image database is very diverse and can be considered as a representation of the content of the Web or generic image collections. In this work, we randomly selected 30 images to be used as queries.

In the experiments we used the descriptors described in Section 3: the three MPEG-7 descriptors (Scalable Color Descriptor (SCD), Color Layout Descriptor (CLD) and Edge Histogram Descriptor (EHD)); the Fuzzy Color and Texture Histogram (FCTH), Color and Edge Directivity Descriptor (CEDD), Local Color Histogram (LCH), Border/Interior Pixel Classification (BIC); and the two descriptors proposed (LCPC and LCPC+Edge).

The MPEG-7 descriptors, FCTH and CEDD are available on [Lux and Chatzichristofis 2008]. An implementation of BIC was made available to us by its authors. Therefore all those descriptors can be used as baselines in our experiments, allowing us to draw a direct comparison to our descriptors.

In order to evaluate the effectiveness of LCPC and LCPC+Edge against the baselines, we use the mean average precision (MAP) and precision-recall metrics [Baeza-Yates and Ribeiro-Neto 1999]. These metrics consider that, for each image query, a set of relevant images has been defined. In the CCD and Wang databases, this set is represented by the ground truth images. For the generic 100k image collection, relevance judgments are accomplished through human evaluation.

The set of relevant images in the $100K$ dataset to each of our 30 image queries is determined by the *pooling* technique, a common method used in TREC collections to assemble the relevance assessments [Hawking et al. 1999]. For each image query we ran LCPC and all the baseline descriptors presented in section 3. The 30 top-ranked images retrieved by each descriptor were put together into an image pool with duplications removed to create a set of unique images relevant for each image query. This makes impossible to know which descriptor retrieved which image.

The images were classified by 18 evaluators (graduate and undergraduate students), instructed to perform a binary judgment and to set an image as relevant or non-relevant with respect to the

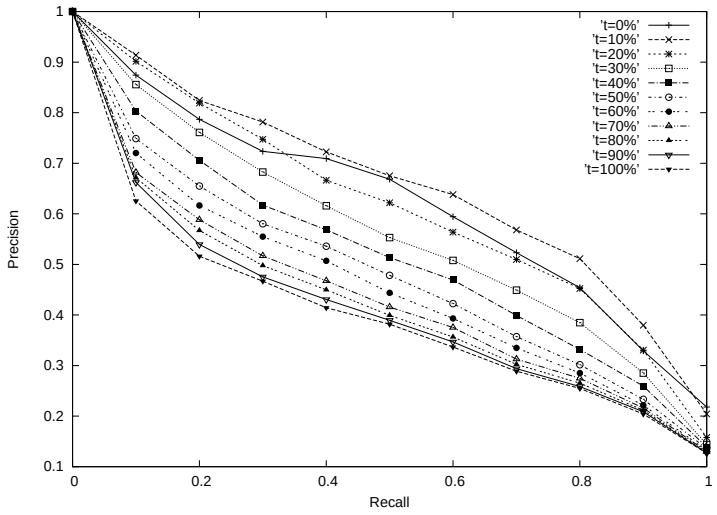



Fig. 3. Precision-recall curves for LCPC + Edge descriptor.

corresponding image query. Each evaluator executed at least 30 queries using one of the image descriptors evaluated. Note that the evaluator has no knowledge regarding the descriptor that he/she used. A few evaluators evaluated more than one descriptor to assure a minimum of 6 evaluations for each of the descriptors.

As a result of this relevance assessment, we have a set of images with each image labeled as relevant or non-relevant, independently of how they were retrieved. By matching this set against each of the image descriptors we can evaluate their effectiveness. It is important to note that this ranking method does not guarantee that every relevant image in the collection is found, a clearly infeasible task given the size of the image database. For this reason, when a descriptor returns all the images in the relevant set, we say that the query has achieved a 100% of relative recall. This pooling method is used in the Web-based collection of TREC [Hawking et al. 1999] and it was also used in [Coelho et al. 2004]. It avoids the need to evaluate the whole collection and guarantees that the user classifying the images has no knowledge about the strategy used to retrieve them, thus providing an impartial relevance judgment [Penatti and da S. Torres 2010b]. In order to perform the experiments using the 100K image database in a systematic way, a tool for automation of comparative CBIR tests was used [Penatti and da S. Torres 2010a]. This tool allows to run experiments with real users and integrates all the steps necessary to perform practical experiments in an organized and standardized way.

To verify the statistical significance of the results we performed a Wilcoxon matched-pairs rank test which is commonly used to validate this sort of experimental results [Wilcoxon 1945]. The Wilcoxon test is applied to compare whether the average difference between two groups is statistically significant.

### 4.2 Combining Color and Shape Features

Before evaluating the effectiveness of the image descriptors used in this work, we experiment different weight values for combining color and shape features in our LCPC+Edge descriptor. For this experiment we use the Wang dataset. The results can be seen in Figure 3.

It can be observed in Figure 3 that the best results were achieved with the configuration of 10% for shape properties and 90% for color properties. It can also be observed that precision level decreases as the weight for the shape properties is increased. Although this experiment indicated a low weight for the shape features, the result is still useful since the answers are significantly modified when compared

Table I. MAP values on Wang and CCD datasets.

| Image Datasets | LCPC+Edge | LCPC | BIC | CEDD | FCTH | CLD | EHD | SCD | LCH |
|---|---|---|---|---|---|---|---|---|---|
| Wang | 0.67 | 0.62 | 0.59 | 0.59 | 0.58 | 0.51 | 0.39 | 0.38 | 0.55 |
| CCD | 0.82 | 0.84 | 0.84 | 0.67 | 0.71 | 0.56 | 0.42 | 0.35 | 0.54 |

to the LCPC with color properties only, as it is discussed in section 4.5.

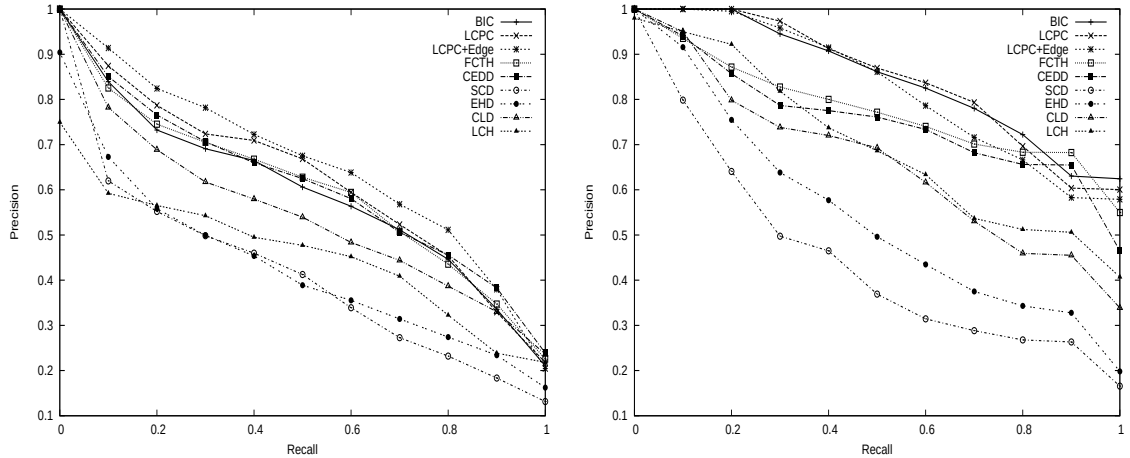
After defining the appropriate combination of color and shape features for the LCPC+Edge descriptor, all the other descriptors (BIC, LCH, LCPC, the MPEG-7 descriptors, FCTH, and CEDD) were tested in the Wang and CCD image databases.

### 4.3 Results in the Wang and CCD Datasets

In this section we present the results of our experiments in the Wang and CCD datasets. Table I presents the mean average precisions (MAP) obtained for LCPC and LCPC+Edge, and the baseline descriptors. We can observe that LCPC outperforms all the baselines, except LCPC+Edge on the Wang database, and the BIC descriptor on the CCD dataset where they were equivalent in terms of MAP. We can also notice that BIC, CEDD and FCTH descriptors had better effectiveness than the MPEG-7 descriptors and LCH in the CCD and Wang datasets. If we compare LCPC to LCPC+Edge, they obtained MAP values very similar in both datasets.

The precision-recall curves obtained from the experiments on the Wang and CCD image databases for all descriptors are shown in Figure 4, where we can see that BIC, LCPC, LCPC+Edge, FCTH, and CEDD presented very close effectiveness in the Wang dataset. We can also observe that the FCTH and CEDD descriptors were better than the MPEG-7 and LCH descriptors in the same dataset. In the CCD database, the descriptors BIC, LCPC and LCPC+Edge remained close, but better than FCTH and CEDD. LCH obtained similar results at recall levels up to 10% and higher precision values at recall levels above 10% and up to 30%. For recall levels above 30%, LCH presented a similar retrieval performance compared to the CLD descriptor.

We performed the Wilcoxon statistical test to verify whether the results achieved by the different descriptors are significant. In our experiments, we considered a difference significant when the significance level is equal to or greater than 95%. To perform the tests, the mean average precision




(a) Wang dataset (b) CCD dataset

Fig. 4. Wang and CCD datasets precision-recall curves.

Table II. Wilcoxon tests for Wang Dataset.

| **Descriptor** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| LCPC | **1** | 0 | 75 | 81 | 30 | 44 | 92 | 99 | 99 | 92 |
| LCPC+Edge | **2** | 75 | 0 | 95 | 81 | 72 | 97 | 100 | 99 | 99 |
| BIC | **3** | 81 | 95 | 0 | 0 | 0 | 84 | 99 | 99 | 81 |
| CEDD | **4** | 30 | 81 | 0 | 0 | 15 | 68 | 100 | 97 | 56 |
| FCTH | **5** | 44 | 72 | 0 | 15 | 0 | 81 | 99 | 97 | 51 |
| CLD | **6** | 92 | 97 | 84 | 68 | 81 | 0 | 97 | 77 | 44 |
| EHD | **7** | 99 | 100 | 99 | 100 | 99 | 97 | 0 | 0 | 98 |
| SCD | **8** | 99 | 99 | 99 | 97 | 97 | 77 | 0 | 0 | 89 |
| LCH | **9** | 92 | 99 | 81 | 56 | 51 | 44 | 98 | 89 | 0 |

Table III. Wilcoxon tests for CCD Dataset.

| **Descriptor** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| LCPC | **1** | 0 | 84 | 76 | 100 | 99 | 100 | 100 | 100 | 100 |
| LCPC+Edge | **2** | 84 | 0 | 15 | 100 | 98 | 100 | 100 | 100 | 100 |
| BIC | **3** | 76 | 15 | 0 | 100 | 100 | 100 | 100 | 100 | 100 |
| CEDD | **4** | 100 | 100 | 100 | 0 | 70 | 98 | 100 | 100 | 100 |
| FCTH | **5** | 99 | 98 | 100 | 70 | 0 | 99 | 100 | 100 | 100 |
| CLD | **6** | 100 | 100 | 100 | 98 | 99 | 0 | 100 | 100 | 40 |
| EHD | **7** | 100 | 100 | 100 | 100 | 100 | 100 | 0 | 65 | 98 |
| SCD | **8** | 100 | 100 | 100 | 100 | 100 | 100 | 65 | 0 | 100 |
| LCH | **9** | 100 | 100 | 100 | 100 | 100 | 40 | 98 | 100 | 0 |

(MAP) of each image query of each method were compared as follows: the MAP of query 1 of the BIC descriptor was compared with the MAP of query 1 for each one of the other descriptors. Similarly, the same procedure is performed to query 2, 3, 4 and so forth until query 50 for the CCD database and until query 10 for the Wang database. The significance level corresponds to a degree of confidence in the significance of the differences in the average precisions obtained by the descriptors. The validation for the Wang database is shown in Table II.

According to Table II, the LCPC+Edge is superior to the BIC descriptor (line 2, column 3) with 95% significance level. It is also significantly superior to all MPEG-7 descriptors. However it is not significantly superior to FCTH e CEDD, given the confidence levels of 72% to FCTH and 81% to CEDD, respectively. This behavior is expected since the image database used is small. Therefore the precision levels are generally high and close to each other, confirming our assumption.

The validation of the results in the CCD database were different from the Wang database as shown in Table III. Here the LCPC, LCPC+Edge and BIC show no significant differences among them. It can be seen in Figure 4 that they present very close precision-recall curves. On the other hand they are significantly superior to all the other descriptors, since they reach more than 95% of confidence level.

### 4.4 Results in the 100K Dataset

In this section we present the results of our experiments[1] using a large image database with over 100,000 images. Table IV presents the mean average precisions (MAP) obtained by LCPC, LCPC+Edge and the baseline descriptors on the 100k image collection.

Note that LCPC obtained significant improvements over the BIC, CEDD, and FCTH descriptors, with 51.85%, 95% and 105% gains in terms of MAP, respectively. It is important to observe that these gain levels were not observed in the smaller datasets (Wang and CCD). This confirms our

[1]The 30 top images retrieved by the better performing descriptors used in our experiments can be viewed at: *http://boiuna.dcc.ufam.edu.br/lcpc-sbbd2011/*

Table IV. MAP and gain precision of the LCPC in the 100K database.

| Descriptor | MAP | Gains of LCPC |
|---|---|---|
| LCPC | 0.41 | - |
| LCPC+Edge | 0.37 | +10.81% |
| BIC | 0.27 | +51.85% |
| CEDD | 0.21 | +95.24% |
| FCTH | 0.20 | +105.00% |
| EHD | 0.12 | +241.66% |
| CLD | 0.11 | +272.72% |
| LCH | 0.08 | +412.50% |
| SCD | 0.07 | +485.71% |

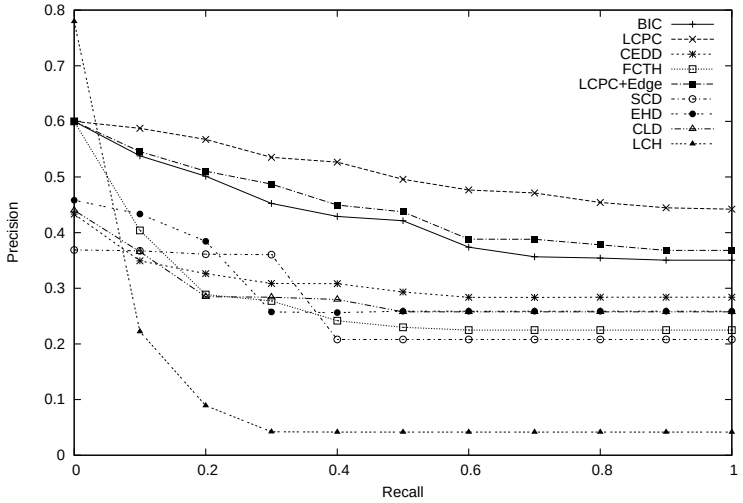



Fig. 5. 100k image dataset precision-recall curves.

assumption that the behavior of descriptors in small and categorized image databases often result in a false sensation of accomplishment given the high precision levels obtained. This result also show that LCPC is more resilient to variations in size, quality, and diversity of the image database. Given that our method also achieved competitive results in the other databases, we conclude that it is an interesting practical solution to be adopted for image retrieval in large and heterogeneous databases.

Figures 5 show the 11-point average precision curves for LCPC, LCPC+Edge and all other descriptors in the 100K image collection. The results presented on that figure corroborates with the assumption that our method is more robust and presents better retrieval effectiveness when applied to the 100K image collection.

In order to make sure that our proposal performs significantly better than the other techniques in the large image database, we also performed the Wilcoxon matched-pairs rank test. The significance level corresponds to a degree of confidence in the hypothesis that two compared results are different, i.e., that the average precisions obtained by two compared methods are really different.

As shown in Table V, the precision levels obtained by LCPC in the 100K image database is significantly higher than BIC, FCTH and CEDD, with a confidence level of 100% proving that LCPC can be deployed in large and heterogeneous image collections. The LCPC is superior to LCPC+Edge with only 94% of confidence, this result show that they are not significantly different in the 100K database. Nevertheless, as discussed earlier, all the answers provided by LCPC have the same colors while the LCPC+Edge is able to return a set of relevant images with different colors, improving the diversity

of the search result.

### 4.5 Discussion

This section discusses the results obtained with some specific image queries submitted to BIC, LCPC, and LCPC+Edge, which achieved the best overall results.

Figure 6 and Figure 7 show the 30 top answers of LCPC and BIC to the image query of a flower. The top left image is the query image used to retrieve the results in each descriptor. This is a typical image in which the separation between foreground and background is crucial for reaching high precision levels. This particular image has distinct color distribution in the foreground and in the background.

Table V. Wilcoxon tests for the 100K image database.

| **Descriptor** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| LCPC | **1** | 0 | 94 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| LCPC+Edge | **2** | 94 | 0 | 70 | 99 | 99 | 100 | 100 | 100 | 100 |
| BIC | **3** | 100 | 70 | 0 | 89 | 99 | 100 | 100 | 100 | 100 |
| CEDD | **4** | 100 | 99 | 89 | 0 | 14 | 100 | 100 | 100 | 100 |
| FCTH | **5** | 100 | 99 | 89 | 14 | 0 | 100 | 100 | 100 | 100 |
| CLD | **6** | 100 | 100 | 100 | 100 | 100 | 0 | 46 | 98 | 100 |
| EHD | **7** | 100 | 100 | 100 | 100 | 100 | 46 | 0 | 86 | 86 |
| SCD | **8** | 100 | 100 | 100 | 100 | 100 | 98 | 86 | 0 | 68 |
| LCH | **9** | 100 | 100 | 100 | 100 | 100 | 100 | 86 | 68 | 0 |

Fig. 6. 30 top images returned by LCPC in the 100K database.

Fig. 7. 30 top images returned by BIC in the 100K database.

As we can see in Figure 6, the LCPC descriptor has outperformed BIC in this query. This is due to the local analysis performed during the feature extraction process, which considers the five partitions as detailed in Section 3.8. This behavior can also be observed in most of the other queries, except those images in which the foreground object (in the central partition) has similar colors to the background objects. In this situation, LCPC and BIC achieved similar levels of precision.

As it could be noted the LCPC+Edge did not outperform the LCPC descriptor. However it did return interesting (perhaps more diverse) results, including in the answer relevant images that have different color when compared to the query image. This situation is shown in Figures 8 and 9, which present the results returned by LCPC+Edge and LCPC respectively, to a query image of a yellow car. Note that LCPC returns yellow cars or other images (possibly irrelevant) also having yellow as the predominant color. The LCPC+Edge descriptor was able to return relevant images with other colors, in this example, images of cars with colors other than yellow. This happens without very large losses in effectiveness when compared to LCPC but with significant gains when compared to the other descriptors.

Note that in Table V the confidence level of the results obtained between LCPC and LCPC+Edge was 94%. Although it is not superior to our threshold of 95% it shows the potential of the combination of color and shape features. It is a strong indication that a thorough investigation on other forms of combining these features or other forms of extracting shape features is worth.

Although it is not the focus of this work we have also performed some performance measures in terms of computing time. More specifically we show the computing time for feature extraction of one image and the calculation of the distance between two images. The measures correspond to the

Fig. 8. 30 top images returned by LCPC+Edge in the 100K database.

Fig. 9. 30 top images returned by LCPC in the 100K database.

Table VI. Average time to extract one feature vector (a) and Average time to compute one distance function between two images (b).

(a)

| Descriptor | Time in ms | Variance |
|---|---|---|
| LCPC | 11 | 0.4 |
| LCPC+Edge | 50 | 2.6 |
| BIC | 10 | 0.4 |
| CEDD | 11 | 0.6 |
| FCTH | 14 | 1.2 |
| EHD | 9 | 0 |
| CLD | 6 | 0 |
| LCH | 8 | 0.6 |
| SCD | 7 | 0 |

(b)

| Descriptor | Time in $\mu$s | Variance |
|---|---|---|
| LCPC | 41 | 3.6 |
| LCPC+Edge | 75 | 3.8 |
| BIC | 32 | 1.6 |
| CEDD | 39 | 3 |
| FCTH | 43 | 2.4 |
| EHD | 31 | 2.4 |
| CLD | 38 | 3.4 |
| LCH | 59 | 1.2 |
| SCD | 32 | 2.8 |

average processing time for 5 query executions. The experiments were run in a dedicated machine for indexing with a Intel Xeon Quadcore processor and 4GB of RAM running Linux Ubuntu 10.04. Despite the computer multi core capability, the implementation of the descriptors were not tunned to parallelization.

It can be observed in Table VI(a) that CLD, SCD, LCH, and EHD presented feature extraction times under 10ms. Interestingly they are exactly the descriptors which presented the worst results in terms of precision. The descriptors BIC, LCPC, and CEDD presented similar processing times, with 10ms, 11ms, and 11ms respectively. FCTH took 14ms which is considerably slower than the others (27% slower than LCPC for instance). As expected LCPC+Edge takes a longer time for feature extraction (50ms) given the complexity of the edge detection algorithm.

Table VI(b) presents the resulting processing time for computing the distance between two images for each descriptor. Note that the processing times are expressed in microseconds ($10^{-6}$s) whereas the feature extraction times are expressed in milliseconds ($10^{-3}$s). Therefore it is clear that any performance problem is usually related to the feature extraction procedure.

From these results we can conclude that the LCPC descriptor presents performance similar to the descriptors BIC, CEDD, and FCTH. However LCPC achieved significant better retrieval effectiveness.

## 5. CONCLUSIONS AND FUTURE WORK

This article presented an evaluation of the retrieval effectiveness of several descriptors on large image databases. Among the image descriptors evaluated there is a new descriptor proposed, namely LCPC, and a combination of LCPC with shape features. Both descriptors are partition-based approach for image retrieval aiming at large and heterogeneous image databases.

In order to improve the effectiveness of the search results, our approach defines partitions with the goal of separating the foreground from the background information available on each image. This allows the extraction of local features, specifically the color distribution and concentration in each individual partition.

The results show that our approach achieves higher precision levels compared to the other descriptors. The gains over the baselines are particularly significant on the large image database rather than on the small image collections adopted in the experiments. These results were confirmed by statistical tests demonstrating significant gains.

This confirms our assumption that in general most image descriptors tend to achieve high precision levels in small databases, whereas in large image databases the precision level tend to be low.

Our future work includes the use of parallelism to extract and calculate the similarity between the

five histograms in order to address efficiency (processing cost) issues. We also plan to investigate in more detail the size and proportion of the foreground and background partitions. The parameters of the edge descriptors and the weight used for combining the color and shape features should also be better exploited in our future work.

## REFERENCES


Azernikov, S. Sweeping solids on manifolds. In *Proceedings of the ACM Symposium on Solid and Physical Modeling*. New York, USA, pp. 249–255, 2008.

Baeza-Yates, R. A. and Ribeiro-Neto, B. A. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

Canny, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8 (6): 679–698, 1986.

Chatzichristofis, S. A. and Boutalis, Y. S. Cedd: Color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In *Proceedings of the International Conference on Computer Vision Systems*. Santorini, Greece, pp. 312–322, 2008.

Cieplinski, L. Mpeg-7 color descriptors and their applications. In *Proceedings of the International Conference on Computer Analysis of Images and Patterns*. Warsaw, Poland, pp. 11–20, 2001.

Coelho, T. A. S., Calado, P. P., Souza, L. V., Ribeiro-Neto, B., and Muntz, R. Image retrieval using multiple evidence ranking. *IEEE Transactions on Knowledge and Data Engineering* 16 (4): 408–417, 2004.

Douze, M., Jégou, H., Sandhawalia, H., Amsaleg, L., and Schmid, C. Evaluation of gist descriptors for web-scale image search. In *Proceedings of the ACM International Conference on Image and Video Retrieval*. Santorini, Greece, pp. 19:1–19:8, 2009.

Hawking, D., Craswell, N., Thistlewaite, P., and Harman, D. Results and challenges in web search evaluation. In *Proceedings of the International Conference on World Wide Web*. Toronto, Canada, pp. 1321–1330, 1999.

Hörster, E., Lienhart, R., and Slaney, M. Image retrieval on large-scale image databases. In *Proceedings of the ACM International Conference on Image and Video Retrieval*. Amsterdam, The Netherlands, pp. 17–24, 2007.

Li, J. and Wang, J. Z. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (9): 1075–1088, 2003.

Lux, M. and Chatzichristofis, S. A. Lire: Lucene image retrieval: an extensible java cbir library. In *Proceeding of the ACM International Conference on Multimedia*. Vancouver, Canada, pp. 1085–1088, 2008.

Mai, F., Hung, Y. S., Zhong, H., and Sze, W. F. A hierarchical approach for fast and robust ellipse extraction. *Journal of Pattern Recognition* 41 (8): 2512–2524, 2008.

Manjunath, B. S., Ohm, J. R., Vasudevan, V. V., and Yamada, A. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology* 11 (6): 703–715, 2001.

Pass, G., Zabih, R., and Miller, J. Comparing images using color coherence vectors. In *Proceedings of the ACM International Conference on Multimedia*. Boston, USA, pp. 65–73, 1996.

Penatti, O. A. B. and da S. Torres, R. Color descriptors for web image retrieval: a comparative study. In *Proceedings of the Brazilian Symposium on Computer Graphics*. Campo Grande, Brazil, pp. 163–170, 2008.

Penatti, O. A. B. and da S. Torres, R. Eva: an evaluation tool for comparing descriptors in content-based image retrieval tasks. In *Proceedings of the ACM Multimedia Information Retrieval*. Philadelphia, USA, pp. 413–416, 2010a.

Penatti, O. A. B. and da S. Torres, R. User-oriented evaluation of color descriptors for web image retrieval. In *Proceedings of the European Conference on Research and Advanced Technology for Digital Libraries*. Glasgow, UK, pp. 486–489, 2010b.

Sikora, T. The MPEG-7 visual standard for content description - An overview. *IEEE Transactions on Circuits and Systems for Video Technology* 11 (6): 696–702, 2001.

Stehling, R. O., Nascimento, M. A., and Falcão, A. X. A compact and efficient image retrieval approach based on border/interior pixel classification. In *Proceedings of the International Conference on Information and Knowledge Engineering*. McLean, USA, pp. 102–109, 2002.

Stricker, M. and Orengo, M. Similarity of color images. In *In Proceedings of Storage and Retrieval for Image and Video Databases*. San Jose, USA, pp. 381 – 392, 1995.

Swain, M. J. and Ballard, D. H. Color indexing. *International Journal of Computer Vision* 7 (1): 11–32, 1991.

Wilcoxon, F. Individual comparisons by ranking methods. *Biometrics Bulletin* 1 (6): 80–83, 1945.

Wong, K. M., Po, L. M., and Cheung, K. W. Dominant color structure descriptor for image retrieval. In *Proceedings of the International Conference on Image Processing*. San Antonio, USA, pp. 365–368, 2007.

Zagoris, K., Boutalis, Y. S., Papamarkos, N., and Chatzichristofis, S. A. Accurate image retrieval based on compact composite descriptors and relevance feedback information. *International Journal of Pattern Recognition and Artificial Intelligence* 24 (1): 207–244, 2010.