Comparative Study of Global Color and Texture Descriptors for Web Image Retrieval

Otávio A. B. Penatti, Eduardo Valle, Ricardo da S. Torres

penatti@ic.unicamp.br, mail@eduardovalle.com, rtorres@ic.unicamp.br
RECOD Lab — Institute of Computing
University of Campinas
Brazil

Abstract

This paper presents a comparative study of color and texture descriptors considering the Web as the environment of use. We take into account the diversity and large-scale aspects of the Web considering a large number of descriptors (24 color and 28 texture descriptors, including both traditional and recently proposed ones). The evaluation is made on two levels: a theoretical analysis in terms of algorithms complexities and an experimental comparison considering efficiency and effectiveness aspects. The experimental comparison contrasts the performances of the descriptors in small-scale datasets and in a large heterogeneous database containing more than 230 thousand images. Although there is a significant correlation between descriptors performances in the two settings, there are notable deviations, which must be taken into account when selecting the descriptors for large-scale tasks. An analysis of the correlation is provided for the best descriptors, which hints at the best opportunities of their use in combination.

Keywords:
comparative study, color descriptors, texture descriptors, web

1. Introduction

This paper presents a comparative study of global color and texture descriptors considering the Web as the environment of use.

In the latest years, the amount of digital images has grown rapidly. Among the main reasons for that, one may mention digital cameras and high-speed Internet connections. Those elements have created a simple way to generate and publish visual content worldwide. That means that a huge amount of visual information becomes available everyday to a growing number of users. Much of that visual information is available on the Web, which has become the largest and most heterogeneous image database so far.

In that scenario, there is a crucial demand for image retrieval systems [1, 2], which could be satisfied by Content-based Image Retrieval (CBIR) systems. In CBIR systems, the *image descriptor* is a very important element. It is responsible for assessing the similarities between images. Descriptors can be classified depending on the image property analyzed, like, for example, color or texture descriptors, that analyze color or texture properties, respectively.

In CBIR systems, the searching process works as follows. The user queries the system, usually, by using a query image. Its properties are extracted and then compared against the properties from the database images, that had been previously extracted. The comparisons are made by computing distance values and those values are used to rank the database images according to their similarities to the query image. The most similar images are finally shown to the user. The image descriptor is involved in this process in the extraction of images properties and in the distances computations. Therefore, it is clear the critical importance of image descriptors for CBIR systems.

It is known that many image descriptors are application dependent, that is, their performances vary from one application to another. Therefore, conducting comparative evaluation of image descriptors considering different environments of use is very important.

Literature presents several comparative studies for color, texture, and shape descriptors. A recent study [3] compares a large number of image descriptors in five different image collections for tasks of classification and image retrieval. Other studies are specific to certain properties: shape descriptors [4, 5, 6, 7, 8], texture descriptors [9, 10, 11], or color descriptors [12, 13, 14]. Surveys and comparisons markedly related to ours can be found in Sections 2.4 and 2.6.

In comparative studies of descriptors, the Web is rarely considered as the environment of use. In general, the amount of descriptors considered is small and the application analyzed is specific. Besides that, asymptotic theoretical analysis and other efficiency considerations are generally not discussed in detail.

Our study has many novel aspects: first, it considers a large number of descriptors: 24 color and 28 texture descriptors, including both traditional and recently proposed ones. Our evaluation is made in two levels: a theoretical analysis of algorithms complexities and an experimental comparison. The experimental comparison is made over specific and heterogeneous collections. Descriptors are analyzed in a Web environment with a database containing more than 230 thousand images with very heterogeneous content. The experimental analysis considers efficiency and effectiveness aspects. The effectiveness evaluation in the Web environment takes into account how much the descriptors agree with the human perception of semantic similarity, by asking a pool of users to annotate the relevance of answers for each query.

Another important aspect of our study is related to scalability and diversity. How does a descriptor perform as the size of the collection increases considerably? And how does the heterogeneity of the collection affect the descriptors effectiveness? Our experiments in the Web environment address both issues.

The large-scale, heterogeneous nature of the Web can benefit from the use of descriptors in combination. Although the combination of descriptors is a complex topic, with many competing techniques, and thus outside the scope of this work, we perform a general analysis of the complementarity of the best descriptors, which should be taken into account when selecting them for combined use.

We concentrate our evaluation on global image descriptors, since local descriptors have a radically different cost $\times$ benefit compromise, especially in the context of information retrieval involving high-level semantic contexts. Local image detectors and descriptors have been extensively surveyed in [15] and [16, 17], respectively.

We have decided not to include shape descriptors in our study, because almost all of them are segmentation dependent. As is known, segmentation is still a hard and extremely application dependent task. Therefore, shape descriptors are still not mature for a heterogeneous environment like the Web. Readers interested in shape descriptors should refer to one of the comparative studies in specific environments [4, 5, 6, 7, 8]. Another less common kind of descriptor, called spatial relationship descriptor, tries to encode the spatial relationships between objects [18]. However, those descriptors also depend on the segmentation of images.

The paper is organized as follows. Section 2 presents the image descriptor definition used throughout this work and taxonomies for color and texture descriptors. Section 3 presents the results of theoretical analysis of the descriptors, discussing the evaluation criteria used and the theoretical comparative tables. Section 4 presents the experimental evaluation, showing the experimental measures adopted, the implementation details of each descriptor, and the results achieved for the specific collections. Section 5 presents the evaluation for the Web environment. In Section 6, we conclude the paper.

2. Descriptors

Both the effectiveness and the efficiency of content-based image retrieval systems are very dependent on the image descriptors that are being used. The image descriptor is responsible for characterizing the image properties and to compute their similarities. In other words, the image descriptor makes it possible to rank images according to their visual properties.

2.1. Definition

The **image descriptor** can be conceptually understood as responsible for quantifying how similar two images are. Formally, an image descriptor D can be defined as a pair (ϵ_D, δ_D) [19], where ϵ_D is a feature-extraction algorithm and δ_D is a function suitable to compare the feature vectors generated:

- ϵ_D encodes image visual properties into feature vectors, as shown in Figure 1. A feature vector contains information related to the image visual properties, like color, texture, shape, and spatial relationship of objects.
- δ_D compares two feature vectors. As shown in Figure 1, given two feature vectors, the function computes a distance or similarity value between these vectors. The distance or similarity between the vectors is considered as the distance or similarity between the images from which the vectors were extracted.

It is worth noting that in some papers from literature, what we call here as *feature vector* is considered the descriptor, with the distance function being accounted elsewhere. We adopt the definition of [19], in which the descriptor also includes the distance function, since feature vectors need a specific distance function to establish the geometry of the description space: the vectors alone, without the metric, are meaningless. That is better understood if we observe that the same type of feature vector may have radically different performances when used with different distance functions.

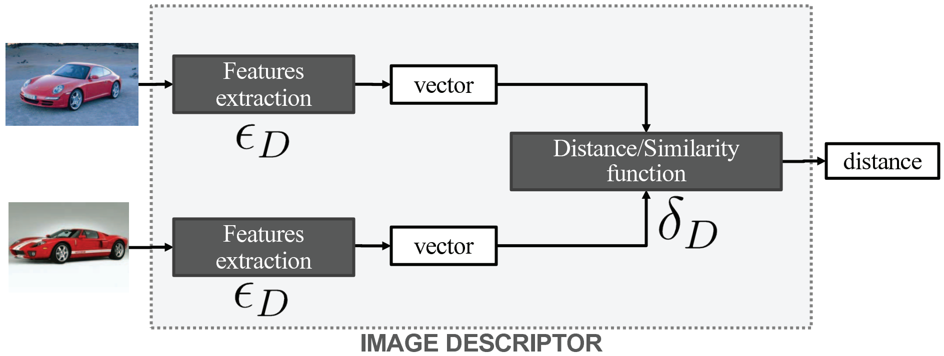


Figure 1: Image descriptor components.

2.2. Distance functions

Distance functions are very important for the image descriptors. Their choice has a huge impact in the descriptor performance.

The most common distance functions are the function L_1, also named Manhattan or city-block, the function L_2, also known as Euclidean distance, and the function L_∞. Those common functions (or variations of them) are largely used. There are also, more complex functions, like the Earth Mover's Distance (EMD) [20].

2.3. Color descriptors

One of the most important visual properties identified by human vision is color, making it one of the most used properties in CBIR systems.

Taxonomy for color descriptors. Literature presents three main approaches for color analysis, as shown in Figure 2.

The global approach considers the image color information globally. As no partitioning or pre-processing stage is applied to the image during features extraction, descriptors from this approach usually have simple and fast algorithms for extracting feature vectors. However, as no information about the spatial distribution of colors is encoded, those descriptors can have little discriminating power. Many descriptors from global approach generate histograms as feature vectors, like, for example, the *global color histogram* [21] and the *cumulative global color histogram* [22].

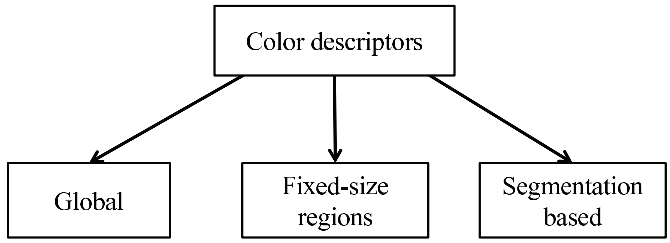


Figure 2: Taxonomy for color descriptors.

The fixed-size regions approach divides an image into cells of fixed size and extracts color information from each cell separately. The descriptors from this approach encode more spatial information, at the cost of usually generating larger feature vectors. Examples of descriptors from fixed-size regions approach are *local color histogram* [21] and *cell/color histogram* [23].

The segmentation-based approach divides an image in regions that may differ in size and quantity from one image to another. This division is usually made by a segmentation or clustering algorithm, what introduces an extra complexity to the features extraction process. Another kind of segmentation is the classification of pixels before feature extraction. Descriptors from that approach usually present better effectiveness, although they are often more complex. Examples of descriptors from segmentation-based approach are *color-based clustering* [24] and *dominant-color* [25, 26].

Authors often do not give their methods neither a name nor an acronym. To refer to those methods less awkwardly through the text, we have taken the liberty of giving them a short descriptive name and acronym. For methods already named in the original publication, we have, of course, used their standard designation.

2.4. Previous comparisons on color descriptors

Besides the studies mentioned in the introduction, studies focused on the MPEG-7 descriptors are especially related to ours, due to the large number of descriptors involved on the standard.

A comparative study of the color descriptors from the MPEG-7 standard [13] shows that CSD [26] has the best performance while CLD [26] is pointed as the most sensitive to noise.

Another comparative study of the MPEG-7 color descriptors [14] points that CSD [26] has the best effectiveness, being better than SCD [26], CLD [26], and DCD [26], in this order. According to the study, DCD, although being the most computationally complex descriptor, yields the worst results since it focuses on parts of images and not on images as a whole.

The use of color in local features is a subject in itself, and outside the scope of this work. In any case, the use of local features often imply computational costs which limits their use in Web-like environments. The interested reader is, nevertheless, encouraged to refer to the comprehensive study of van de Sande et al. [17].

2.5. Texture descriptors

Texture is an important property for the characterization and recognition of images. This fact is observed by the great amount of research involving texture analysis of images [27, 28, 29, 10].

Texture is easily recognized in images, as can be shown by elements like sand, leaves, clouds, bricks, etc. However, it is difficult to provide a formal definition for it. Literature gives a variety of definitions, as shown in [27]: "its structure is simply attributed to the repetitive patterns in which elements or primitives are arranged according to a placement rule"; "an image texture is described by the number and types of its primitives and the spatial organization or layout of its primitives". According to [30], texture can be described by spatial, frequency or perceptual properties. In a general way, texture can be understood as a set of intensity variations that follow certain repetitive patterns.

Differently from color, texture is difficult to be analyzed considering the value of a single pixel, since it occurs mainly due to the variation in a neighborhood of pixels. That makes it possible to name some attributes for textures. According to [31], texture has attributes of roughness, contrast, directionality, regularity, coarseness, and line likelihood, the first three ones being the most important.

Although most texture descriptors work on gray-scale, only few of them specify how color images should be converted in order to optimize the descriptor performance.

Taxonomy for texture descriptors. There are several approaches for texture extraction [30, 27]. The taxonomy presented in Figure 3 is the one found in [27].

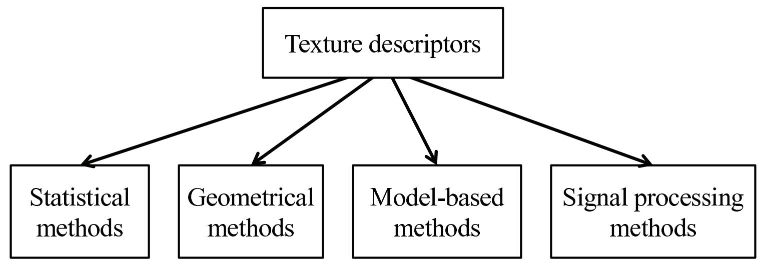


Figure 3: Taxonomy for texture descriptors [27].

One of the most traditional ways to analyze the spatial distribution of the gray levels of an image is by statistical analysis, as, for example, by computing the probability of co-occurrence of gray values in different distances and orientations. The statistics can be computed over the values of single pixels (first-order statistics) or over the values of pairs of pixels (second-order statistics) [27]. Methods that characterize textures by means of histograms also have statistical information about texture. One of the most popular statistical methods is the *co-occurrence matrix* [32].

Geometrical methods analyze textures by "texture elements" or primitives. This analysis is made considering the geometrical properties of the primitives, like size, shape, area, and length. Having the primitives being identified in an image, placement rules are extracted from them, like grids or statistics from relative vectors that join the primitives centroids [33]. That kind of analysis becomes difficult for natural textures, because the primitives and the placement rules can be irregular. For example, describing a wall of bricks by the primitive *brick* and a *grid* as placement rule can be simple. However, describing the clouds in the sky is much more difficult, since the element *cloud* can have variable size and shape and the positioning of them is more complex.

Model-based methods rely on the construction of image models that can be used to describe and synthesize textures. The parameters from the model capture the essential perceived qualities of texture [27]. For example, texture elements can be modeled as: a dark or a bright spot, an horizontal or vertical transition, corners, lines, etc. Descriptors from that approach work well for regular textures. The *local binary pattern* [34] descriptor is an example of model-based descriptor.

Signal processing methods characterize textures by applying filters over the image. Both spatial domain and frequency domain filters can be used. Descriptors based on wavelets and Gabor filters follow that approach, like the *homogeneous texture descriptor* [35, 28], for instance.

We have addressed the issue of methods originally without a name or acronym in the same way as we did for the color descriptors (Section 2.3).

2.6. Previous comparisons on texture descriptors

Literature has some comparative studies on texture descriptors [9, 11, 10, 36]. A study compares a Fourier-based descriptor with a Gabor-based descriptor [36] trying to identify the descriptor that is the most robust to noise. The results show that the Fourier-based descriptor has better performance in images with no noise and that the Gabor-based descriptor is better for noisy images.

Another study compares the texture descriptors of MPEG-7 standard [10]. Considering features extraction cost, the study shows that TBD [26, 35] is the most expensive and that EHD [26] is the cheapest one. Besides that, the study points out that HTD [35, 28] captures global information, while TBD captures global and local information and EHD captures only local information. The study also shows that TBD is not indicated for CBIR tasks, being more useful for image browsing and for defects detection. Additionally, HTD is suggested for being used in CBIR and in texture segmentation tasks, while EHD is recommended for CBIR tasks. The study also shows that HTD and TBD are less sensitive to noise while EHD is not recommended in environments with noise. The experiments realized by the study [10] show that TBD has low effectiveness and that HTD is not good for images with rotation. That last problem of HTD is noticed in the literature, given the number of versions proposing HTD with rotation invariance, like the descriptors HTDR [37], HTDI [38], and Han Gabor [39] present in our study.

3. Theoretical comparison

The comparative study performed in this paper comprises two analyses. This section presents the first one: a theoretical evaluation conducted for 24 color descriptors and for 28 texture descriptors. The theoretical evaluation considers the asymptotic complexities of the descriptors algorithms. The next section presents the second analysis: an experimental evaluation for the most promising color and texture descriptors. In the following section, color and texture descriptors are tested in a Web-like environment.

3.1. Evaluation criteria

Based on the main elements related to the search process in a content-based image retrieval system (as explained in Section 1), the following criteria are used to compare the descriptors: features extraction complexity, distance function complexity, storage requirements, effectiveness, and validation environment.

Features extraction complexity. The features extraction algorithm of a descriptor is used whenever the features from an image need to be extracted. The extraction is required basically in two moments in a Web CBIR system. The first one is when the images are being collected from the Web to be included in the local database, which is usually an *off-line* process. The second moment is when a query is performed, which is an *on-line* process. The on-line part directly affects the response time of the system. Therefore, it is important for the feature extraction algorithm to be fast.

Our complexity analysis is informed in asymptotic big-O notation, which ignores additive and multiplicative constants. As such, it should be taken with a grain of salt, since two $O(n)$ algorithms, though theoretically similar in behavior, may vary considerably in practice, because of the hidden constants.

It will be seen that almost all extractors are linear, or at worse (when they work on the frequency domain), log-linear in the number of image pixels. There are notably exceptions, though, for example: methods based on scale-spaces must take into account the number of scales analyzed. The rare methods which perform complex adaptive clustering are more expensive.

Distance function complexity. The distance function of a descriptor is very time consuming when a query is being processed in a CBIR system. Distance functions need to be fast, because during the search process, the query image will be compared to a large number of candidate images.

The distance function is also important for indexing issues. The use of indexing structures is important for systems with large databases and, therefore, critical in a Web environment. Without indexing structures the response time would be unacceptable. Indexing, however, imposes restrictions on distance functions, because simple, axis-monotone norm-based functions are much friendlier to indexing structures than elaborated and unpredictable decision procedures. The interaction between descriptors and indexing structures is, however, complex, depending on the statistical characteristics of the feature vector, the image dataset, and the properties of the index employed. Prospective studies on that subject may be found in [40, 41].

Again, in our theoretical analysis, the complexity is given asymptotically. It will be seen that almost all methods are based on simple distances (Manhattan, Euclidean, etc.) which are linear in the feature vector size.

Storage requirements. The image descriptor stores into feature vectors the encoded image properties. Each image managed by a CBIR system is associated with one or more feature vectors. As a result, the storage space required for feature vectors is proportional to the amount of images in the database. In a Web scenario, in addition to the very large database size, there is the issue of image heterogeneity, making almost indispensable to employ several descriptor at once. As a consequence, the storage space required is also proportional to the array of descriptors employed.

Seldom the feature vector size for a method is fixed, more usually it depends on a set of parameters which can be adapted from application to application. For color descriptors, for example, many descriptors allow to customize how many colors are to be chosen on a quantization step. Methods based on frequency-domain transforms (Wavelets, Fourier, DCT) often have a choice of selecting how many transform coefficients are to be considered. Many methods can be seen as multidimensional histograms, and, as such, grow as fast as the product of the number of bins in each dimension. Our analysis considers those mentioned possible variations in vectors sizes.

Dimensionality reduction techniques may be employed in order to alleviate the storage requirements of the larger descriptors. However, often (but not always) they also impose a loss in effectiveness. There are many dimensionality reduction techniques, including the linear projection techniques like Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), and metric embedding techniques (both linear and non-linear). The interaction between dimensionality reduction and descriptor performance are, however, far from trivial and outside the scope of this work. Here we are interested in evaluating the "raw" behaviour of the descriptors, with minimum external influences.

Effectiveness. Effectiveness stands for the capacity of a descriptor to retrieve relevant images. An effective descriptor puts together images which are related in the users' viewpoint, so that during retrieval relevant images are ranked first. The success of an image retrieval system is, of course, directly dependent on the quality of its results, an thus, on the effectiveness of the descriptors. A user might tolerate occasional delays, but not systematically wrong answers.

Measuring effectiveness. Effectiveness poses an issue: it depends upon the agreement about which answers are to be considered relevant for queries. Reaching that agreement may be very difficult, since the concept of relevant is dependent on each user mental model of the query.

In practice, experimenters have two choices. First, to use a categorized image database, where the database categorization is considered the *ground truth* and creates the possibility for an automatic effectiveness evaluation. A second way to evaluate descriptors is conducted over databases with no fixed ground truth. In those cases, the system performs a pool of queries and selected users judge how well the system fared according to their own subjective criteria. That procedure introduces a source of variability, but has the advantage of evaluating the system according to the opinion of potential users.

Once the matter of annotation is settled, there remains the choice of which metric to use to summarize effectiveness over the set of queries (and occasionally users) employed. The most traditional choice are the measures of Precision and Recall, often in combination or as a function of each other (in a Precision $\times$ Recall graph, for example). Precision measures the fraction of relevant images in proportion to the answer set, while Recall measures the fraction of relevant images retrieved in the answer set in proportion to all relevant images existing in the database. A perfect system would have a Precision of 1 (all images in the answer set are relevant) and a Recall also of 1 (all relevant images in the database are retrieved in the answer set). In practice there is a compromise between the two measures: Recall is related to the concept of True Positive Rate, of the theory of receiver operating characteristic (ROC), while Precision is related to $1-$False Positive Rate. Because of the compromise between Precision and Recall, sometimes a combination of the two is employed, as a single metric. This is the case of the mAP (mean average precision), F1-score, etc.

The problem of Precision and Recall (and their combined measures) is that they tend to ignore the ranking of the images. For example a Precision of 0.5 indicates that half the images in the answer set are correct without saying if it is the first half or the second half. In order to count that, those measures can be taken at specific points, considering answer sets of (usually small) growing sizes. Comparing the Precision at an answer set of 10, 20 and 30 images (called P_{10}, P_{20} and P_{30}, respectively), one can have a better idea of how the correct images are ranked. Usually, one is interested only in the top 10 or 20 answers, corresponding to the first page of answers, since it is observed in real-world search engines that users seldom take the effort of checking the second page, instead preferring to reformulate the query. Corresponding measures for the Recall (R_{10}, R_{20}, etc.) also exist.

Measures that incorporate the concept of ranking *ab initio* have also been proposed, notably for the evaluation of the descriptors on the MPEG-7 (the Rank itself, AVR, MRR, NMRR, ANMRR, etc. [26]). As it will be seen, there is less consensus on the literature on those measures and more tendency of authors on making *ad hoc* adjustments each time they are used, instead of keeping them consistent.

Validation environment. The validation environment comprised, in addition to the chosen metrics of efficiency and effectiveness, the image databases in which the descriptor was tested, and, most importantly, other descriptors to which it was compared. Those aspects reveal, for example, if the descriptor was adequately assessed.

It will be seen that there is little standardization in the validation environment, specially for color descriptors. Several different evaluation measures and image databases are employed, making it difficult to make a meta-analysis. For texture descriptors, validation is commonly based on classification experimental designs, instead of retrieval ones. In general, it will be seen little interest in measuring efficiency.

3.2. Analysis for color descriptors

Table 1 presents the asymptotic complexities of features extraction algorithms and distance functions, the feature vectors size, and the taxonomy class of each descriptor. Descriptors are sorted, inside each taxonomy class, in reverse chronological order.

The great majority of color descriptors extracts the feature vector in linear time on the number of pixels. However, there are some descriptors, like DCSD [42], DCDI [43], DCD [26], WC [44], and CBC [24], that have more complex extraction algorithms. It is observed that many of the more complex descriptors are based on segmentation techniques.

Considering the distance functions complexity, it is observed that the great majority of descriptors is linear to the feature vector size. Notwithstanding, descriptors like DCDI [43], DCD [26], SBS [45], and CBC [24] are more complex for distance computations.

For the storage requirements criterium, we highlight the descriptors CW-LUV [46] and CW-HSV [47] because of the compact feature vectors they generate.

Table 2 shows the validation environment of each descriptor. Many of the color descriptors are validated using simple image databases and the databases are usually small, rarely having more than 10 thousand images. One of the image databases commonly used is Corel, in whole or in parts, but it is often used in non-standard subsets. Table 2 also shows that some descriptors are not compared with other descriptors. The *global color histogram* (GCH) comes closer to a standard method again which most other methods tend to compare, but this is not an absolute rule – some methods for example, compare only with versions of themselves.

The most common evaluation measures are Precision, Recall, and ANMRR. There is seemingly little interest in measuring efficiency: retrieval and extraction times are rarely offered, and no formal comparison of the complexities is ever considered.

Table 1: Color descriptors: reference, short description, and complexity analysis.

Class	Ref.	Acronym[1]	Short description	Complexity[2] Extraction	Complexity[2] Distance	Vector Size	Notes
Global	Williams and Yoon [48]	JAC	Joint autocorrelogram of color, texture, gradient and rank	$O(n)$	$O(vs)$	$N_C \times N_{Grad} \times N_{Rank} \times N_{Text} \times N_{Dist}$ values	N_{Grad}, N_{Rank} e N_{Text} = number of quantization levels for gradient magnitude, rank, and texturedness, respectively
	Nallaperumal et al. [46]	CW-LUV	Wavelet coefficients of color histogram in CIELuv space	$O(n)$	$O(vs)$	127 bits	
	Utenpattanant et al. [47]	CW-HSV	Wavelet coefficients of color histogram in HSV space	$O(n)$	$O(vs)$	63 bits	
	Chaira and Ray [49]	Chaira Fuzzy	Color histogram with fuzzy distance based on gamma membership function	$O(n)$	$O(vs)$	$3 \times N_C$ values	
	Moghaddam et al. [44]	WC	Color correlogram based on directional wavelet coefficients	$O(n \log n)$	$O(vs)$	$N_{Wav} \times N_{Dist} \times N_{Mat} \times N_S$ values	N_{Wav} = wavelet coefficients quantization, N_{Mat} = number of matrices, N_S = number of scales
	Paschos et al. [50]	CM*	Spectral chrominance of an image by moments of chromaticity, based on chromaticity histogram	$O(n) + O(N_C^2 \times N_{Mo})$	$O(vs)$	$2 \times N_{Mo}$ values	N_{Mo} = number of moments
	Manjunath et al. [26] - (Section B)	SCD*	Haar transform of color histogram	$O(n)$	$O(vs)$	N_{Haar} bins	N_{Haar} = number of Haar coefficients
	Manjunath et al. [26] - (Section C)	CSD*	Color histogram in the perceptual HMMD space, computed with a scale sensitive structuring window	$O(n)$	$O(vs)$	32 to 184 values of 8 bits each	
	Huang et al. [51]	ACC	Color autocorrelogram	$O(n)$	$O(vs)$	$N_C \times N_{Dist}$ values	
	Stricker and Orengo [22]	CGCH	Cumulative color histogram	$O(n)$	$O(vs)$	N_C values	
	Swain and Ballard [21]	GCH	Global color histogram	$O(n)$	$O(vs)$	N_C values	
Global and Fixed-size regions	Lu and Chang [52]	Color Bitmap	Global and local information by comparing average color of blocks to the whole image average.	$O(n)$	$O(vs)$	$3 \times N_B$ bits + 6 values	

...continues on next page...

[1] Names originally given by the authors are marked with *.

[2] n = number of pixels in an image, vs = vector size, N_C = number of colors in quantized space, N_D = number of dominant colors, N_{Dist} = number of distances, N_B = number of blocks or cells, N_R = number of regions.

...Table 1: continued from previous page...

Class	Ref.	Acronym[1]	Short description	Complexity[2]		Vector Size	Notes
				Extraction	Distance		
Fixed-size regions	Sun et al. [53]	CDE*	Spatial information of colors, using entropy	$O(n)$	$O(vs)$	$2 \times N_C$ values	
	Stehling et al. [23]	CCH*	For each color in image, histogram representing its distribution among the cells	$O(n)$	$O(vs)$	$0.45 \times N_C \times N_B$ or $N_C \times N_B$ values †	† average case or worst case, respectively
	Li [45]	SBS*	Color histograms for each block in an image combined by weighting according to their position	$O(n)$	$O(k \times m \times N_C)$	N_B or $N_C \times N_B$ values †	† value or histogram for each block; k = N_B in query image, m = N_B in database image
	Manjunath et al. [26] - (Section E)	CLD*	Uses DCT on average colors of an image divided by a grid	$O(n \log n)$	$O(vs)$	$3 \times N_{DCT}$ values	N_{DCT} = number of DCT coefficients
	Swain and Ballard [21]	LCH	Concatenation of color histograms, obtained by dividing image on a grid	$O(n)$	$O(vs)$	$N_C \times N_B$ values	
Segmentation based	Yang et al. [43]	DCDI	DCD (see below) with a new color clustering algorithm	$O(n)$ + †	$O(N_{Dq} \times N_{Di})$	$4 \times N_D$ values	† Linear Block Algorithm (LBA) has unknown complexity, but lesser than GLA; $N_{Dq} = N_D$ in query image, $N_{Di} = N_D$ in database image
	Wong et al. [42]	DCSD*	Combination of DCD (see below) and CSD (see above)	$O(n)$+note[3]	$O(vs)$	$3 \times N_D$ values	
	Lee et al. [54]	Lee SC	Color spatial information with color adjacency histogram and color vector angle histogram	$O(n)$	$O(N_C)$	$2 \times N_C + N_{Bord}$	N_{Bord} = number of border pixels
	Stehling et al. [55]	BIC*	Two color histograms in RGB space, one for border pixels, one for interior pixels	$O(n)$	$O(vs)$	$2 \times N_C$ values	
	Manjunath et al. [26] - (Section D)	DCD*	Dominant colors (most representative colors), as well as spatial coherence and color variances	$O(n)$+note[3]	$O(N_{Dq} \times N_{Di})$	$4 \times N_D$ values	$N_{Dq} = N_D$ in query image, $N_{Di} = N_D$ in database image
	Stehling et al. [24]	CBC*	Characterization for each region of an image, which are determined by color clustering	$O(n \log n)$	$O(N_{Rq} \times N_{Ri} \times \log(N_{Rq} \times N_{Ri}))$	$6 \times N_R$ values	$N_{Rq} = N_R$ in query image, $N_{Ri} = N_R$ in database image
	Pass et al. [56]	CCV*	Two color histograms in RGB space, one for large regions, one for small regions	$O(n)$	$O(vs)$	$2 \times N_C$ values	

[3] Employs the Generalized Lloyd Algorithm (GLA), commonly used in the *k-means* clustering algorithm, whose complexity is currently unknown.

Table 2: Color descriptors: validation environment.

Descriptor	**Collection size**	**Collection**[4]	**Evaluation measures**[5]	**Compared with**[5]
JAC [48]	24 000	Berkeley Digital Library	*Scope*×Recall[6]	GCH, *Joint Histogram, Color correlogram*
CW-LUV [46]	100	N/A	*ANMRR, AVR, MRR, NMRR*	HSV and CIELuv color spaces
CW-HSV [47]	2 997	Heterogeneous from the Web	*ANMRR*	Itself
Chaira Fuzzy [49]	(1) 100, (2) 120	(1) Country flags, (2) Textures	Precision, Recall, Precision×Recall	3 different distance functions: *fuzzy divergence, fuzzy similarity measure, GTI model*
WC [44]	1 000	COREL	Precision, *Rank*	*Simplicity, WBIIS, Color correlogram, Border correlogram*
CM [50]	2 266	VisTex (smooth, noise, etc)	Precision, Recall	Histograms intersection
SCD [26]	5 000	CCD MPEG-7	*ANMRR*	Itself
CSD [26]	5 000	CCD MPEG-7	*ANMRR*	Itself
ACC [51]	14 554	Heterogeneous	*r-measure, avg-r-measure, p1-measure, avg-p1-measure*; Recall×*Scope*	GCH, CCV, *CCV/C*
CGCH [22]	3 000	Heterogeneous	*Rank*	GCH
GCH [21]	N/A	Objects in close	*Average match percentile*	*Histogram intersection, Incremental intersection*
Color Bitmap [52]	(1) 800, (2) 470, (3) 10 000	(1) Animations, (2–3) Full color	*Retrieval accuracy*	GCH, *Color moments, Chang and Liu method*
CDE [53]	8 000	Heterogeneous from the Web	*ANMRR*, Precision, Recall	*SCH, Geostat*
CCH [23]	20 000	COREL	Precision×Recall, θ_{abs}, θ_{rel}	GCH, CCV, LCH
SBS [45]	N/A	N/A	N/A	GCH
CLD [26]	5 000	CCD MPEG-7	*ANMRR*	Itself
LCH [21]	N/A	Objects in close	*Average match percentile*	*Histogram intersection, Incremental intersection*
DCDI [43]	2 100	COREL	*ARR, ANMRR*	DCD
DCSD [42]	5 466	CCD MPEG-7	*ANMRR, NMRR*	DCD, CSD, SCD, CLD
Lee SC [54]	5 000	Heterogeneous	*ANMRR*, Precision, Recall	GCH, *Hybrid Graph Representation, Color correlogram*
BIC [55]	20 000	COREL	Precision×Recall, θ×Recall, P_R, P_{30}, R_{30}, P_{100}, R_{100}, *3P-Precision, 11P-Precision*	GCH, CCV, CBC, LCH
DCD [26]	5 000	CCD MPEG-7	*ANMRR*	Itself
CBC [24]	20 000	COREL	Precision × Recall, *NavgR*	CCV, *CMM*, GCH, LCH, *CSH*
CCV [56]	(1) 11 667, (2) 1 440, (3) 1 005, (4) 93, (5) 349	(1) Chabot, (2) QBIC, (3) COREL, (4) PhotoCD, (5) Video scenes	*Rank*	GCH

3.3. Analysis for texture descriptors

Table 3 presents the results of the theoretical analysis of texture descriptors in terms of algorithms complexity for features extraction and distance

[4]*Heterogeneous* and *Heterogeneous from the Web* mean that a non-standard heterogeneous collection was employed; in the latter case the images were collected from the Web.

[5]Evaluation measures and descriptors names that appear in italic are not used in this study, but their explanations can be found in the papers where they are mentioned.

[6]We make a distinction between metrics which appear isolated and metrics which appear *versus* another. The latter appear with a '×' between them.

computation. Table 3 also shows the feature vectors sizes and presents the taxonomy class of each texture descriptor. Descriptors are sorted in reverse chronological order inside each taxonomy class.

From Table 3, it is noticed that several descriptors present features extraction complexity equal to $O(n \log n)$. Most of these more complex descriptors are based on image filtering algorithms, using Gabor filters in many cases.

Considering the complexities of distance functions, it is predominant the use of linear distance functions, which shows that the most laborious step related to texture description is the features extraction phase.

From the analysis of feature vectors sizes, we highlight the compact feature vectors generated by the descriptors TBDI [57], TBD [26, 35], EHD [26], and LBP [34].

Table 4 shows the validation environments of each texture descriptor. Considering the databases used for validation, there is a predominance in the use of Brodatz database [58], which indicates a certain standardization in the validation of texture descriptors. However, some works use only part of the Brodatz database instead of using the complete collection. It is also noteworthy that because of the nature of the Brodatz textures, many articles use classification experimental designs instead of retrieval ones. Few descriptors are validated in heterogeneous databases, showing the lack of validation of texture descriptors for general image retrieval tasks.

Considering the descriptors used as baselines for comparisons, there is a clear predominance in the use of descriptors based on Gabor filters. There are also descriptors that are compared with no other descriptor or only compared with variations of themselves.

The evaluation measures most used are related to average retrieval rate. There are some works worried about extraction and retrieval times, but they are still rare.

Table 3: Texture descriptors: reference, short description, and complexity analysis.

Class	Ref.	Acronym[1]	Short description	Complexity[7] Extraction	Complexity[7] Distance	Vector Size	Notes
Statistical	Kiranyaz et al. [59]	2DWAH*	Local directional 2D histograms on the edge field of a scale-space representation, metaphor of a "walking ant"	†	$O(vs)$	$2 \times LS^2$ values	† Several complex steps, including a at least $O(n \log n)$ filter. The description is insufficient to deduce the complexity. LS = line of sight of the ant
	Huang and Liu [60]	QCCH*	Quantized histogram of the compound rate of change of gray values for each pixel in four directions	$O(n)$	$O(vs)$	N_Q values	
	Hadjidemetriou et al. [61]	MRH	Set of gray-scale histograms built on a scale-space representation	$O(n \times \lambda)$	$O(vs)$	$O(N_Q \times L)$	λ = width of separable Gaussian filter, L = number of scales = $\log_2 \sqrt{n} - 1$
	Çarkacioglu and Yarman-Vural [62, 63]	SASI*	Statistics of clique auto-correlation coefficients calculated over structuring windows	$O(S \times L \times n)$	$O(vs)$	$2 \times N_K \times \sum(\lceil W_i/4 \rceil + 1)$ values	$S \times L$ = windows sizes in pixels, W_i = width of ith window
	Zhou et al. [64]	SOH*	Histograms of scale and orientation built using the frequency domain	$O(n)$ or $O(n^2)$ †	$O(t \times \theta \times vs)$	$N_S \times N_K$ values	† for N_S small or for N_S large, respectively; t = number of translations, θ = number of rotations
	Manjunath et al. [26] - (Section IV.C)	EHD*	Edge histograms in different directions by dividing the image into blocks and using edge detectors	$O(n)$	$O(vs)$	$3 \times N_B \times N_{BQ}$ bits	N_{BQ} = borders quantization
	Tao and Dickinson [65]	LAS*	Local activity spectrum from histogram of image gradient field	$O(n)$	$O(vs)$	N_Q values	
	Kovalev and Volmer [66]	CCOM	Co-occurrence matrix considering color information	$O(n)$	$O(vs)$	$2 \times N_Q^2 \times N_{Dist}$ values †	† worst case
	Unser [67]	Unser	Sum and difference histograms based on the sum and difference of random variables of same variance	$O(n)$ †	N/A	$4 \times N_{Dir} \times N_Q$ values	† $O(N_Q)$ for extracting histograms information; N_{Dir} = number of directions
	Haralick et al. [32]	GCOM	Gray level co-occurrence matrix	$O(n)$ †	N/A	$O(N_Q^2 \times N_{Dist} \times N_{Dir})$ or $14 \times N_{Dir}$ values ‡	† $O(N_Q^3)$ for extracting matrices information; ‡ matrix space or matrix information space, respectively; N_{Dir} = number of directions
Model-based and Statistical	Takala et al. [68]	Takala LBP	Local binary patterns (see below) computed on overlapping blocks of the image	$O(n)$	$O(vs_q \times vs_i)$	$O(N_B \times S_{LBP})$	$vs_q = vs$ of query image, $vs_i = vs$ of database image, S_{LBP} = size of LBP employed
	Ojala et al. [34]	LBP*	Gray-scale and rotational invariant local binary patterns	$O(n)$	$O(vs)$	$2 + P$ values	P = number of neighbors

...continues on next page...

[7] n = number of pixels in image, vs = vector size, N_Q = number of colors/values/gray levels in quantized space, N_{Dist} = number of distances, N_B = number of blocks or cells, N_S = number of scales, N_K = number of orientations, N_{WD} = number of wavelet decompositions.

...Table 3: continued from previous page...

Class	Ref.	Acronym[1]	Short description	Complexity[7] Extraction	Distance	Vector Size	Notes
Signal-Processing and Statistical	Janney and Yu [69]	IFLT*	Rotation invariant features obtained from intensity vector derived from texture patch normalized and Haar wavelet filtered	$O(n)$	$O(vs)$	$N_S \times N_Q$ values	
Signal-Processing, Model-based and Statistical	Zegarra et al. [70]	SBPH*	Global texture characterization by using Steerable Pyramid decomposition and local arrangements of texture determined by local binary patterns	$O(n \log n)$	$O(vs)$	$256 \times N_S \times N_K$ or $59 \times N_S \times N_K$ values †	† NU-SBP or U-SBP respectively
Proc	Han and Ma [39]	Han Gabor	Rotation and scale invariant representation by summations on the conventional Gabor filter responses	$O(n \log n)$	$O(vs)$	$2 \times N_S$ or $2 \times N_K$ values †	† scale or orientation invariance, respectively
	Zhang et al. [38]	HTDI	Rotation invariant descriptor based on HTD-MPEG7 by shifting circularly the feature vector based on the dominant orientation	$O(n \log n)$	†	$2 + (2 \times N_S \times N_K)$ values	† used SVM with RBF kernel
	Zegarra et al. [71]	M-SPyd*	Rotation and scale invariant representation based on Steerable Pyramid decomposition	$O(n \log n)$	$O(vs)$	$2 \times N_S \times N_K$ values	
	Huang et al. [72]	CSG*	Composite gradient vectors obtained from sub-images through a wavelet decomposition	$O(n)$	$O(vs)$	$1440/N_K$ values	
	Lee and Chen [57]	TBDI	Efficient method for the original TBD-MPEG7 by using Fourier and Hough transform	$O(n \log n)$	$O(vs)$	13 bits	
	Wang et al. [73]	BSpline	Over complete B-spline wavelet based statistical features and fractal signatures	$O(n)$	$O(vs)$	$20 \times N_{WD}$ values	
	Kokare et al. [74]	Kokare CW	Cosine-modulated wavelet transform	$O(n)$	$O(vs)$	$8 \times N_{WD}$ values	
	Sim et al. [75]	Sim Zernike	Zernike moments computed from power spectrum of texture image	$O(n \log n)$	$O(vs)$	$2 + N_{Mo}$ values	N_{Mo} = number of moments
	Yang and Liu [76]	MERF*	Random field built upon multi-resolution filters with the maximum entropy method	$O(n \log n)$	$O(vs)$	$2 \times (N_S \times N_K \times S_G)$ values	S_G = size of Gabor filter bank
	Sim et al. [77]	Sim DCT	Texture description based on discrete cosine transform	$O(n \log n)$	$O(vs)$	52 values	
	Ro and Kang [37]	HTDR	Feature vector shifts for rotation invariance to the original HTD-MPEG7	$O(n \log n)$	$O(vs)$	$2 \times N_S \times N_K$ values	
	Manjunath et al. [26, 35]	TBD*	Texture's regularity, directionality, and coarseness by analyzing responses from a set of scale and orientation selective filters	$O(n \log n)$	$O(vs)$	12 bits	
	Manjunath et al. [35, 28]	HTD*	Gabors filters in different scales and orientations	$O(n \log n)$	$O(vs)$	$2 \times N_S \times N_K$ values	
	Rubner and Tomasi [78]	Rubner EMD	Clustering with kd-trees over a space based on a set of Gabor filter responses; Distance measured by EMD	$O(n \log n)$ + †	$O(vs_q \times vs_i)$	$N_{Clu} \times (N_K \times N_S + 1)$	†Clustering algorithm of unspecified complexity; $vs_q = vs$ of query image, $vs_i = vs$ of database image, N_{Clu} = number of clusters

Table 4: Texture descriptors: validation environment.

Descriptor	Collection size	Collection[4]	Evaluation measures[5]	Compared with[5]
2DWAH [59]	1 760	Brodatz	*NMRR, ANMRR*	EHD, *EPNH, ARP*; Gabor, *GLCM, ORDC*
QCCH [60]	800	Heterogeneous from the Web and categorized	Precision, P_{20}, P_{50}, P_{80}, Extraction time (seconds)	*SCH*, Gabor
MRH [61]	(1) 108, (2) 91, (3) 8 046	(1) Synthetic, (2) Brodatz, (3) CureT	*Classification accuracy*	*Fourier Spectrum, Color features, Wavelet package*, Co-occurrence matrix, *Markov random fields*
SASI [62, 63]	(1) 1 792, (2) 976, (3) 480, (4) 2 672, (5) 5 920	(1) Brodatz, (2) CUReT, (3) PhoTex, (4) VisTex, (5) All together	*Top-15 average retrieval rate*	Gabor
SOH [64]	600	Brodatz	*Average retrieval rate*	*Wold, MRSAR*
EHD [26]	11 000	MPEG-7	*ANMRR*	Itself
LAS [65]	(1) 1 920, (2) 100	Brodatz and satellite images; more than one texture per image	*Classification accuracy*, Showed query results.	*Gradient indexing (4 different gradient operators: Sobel, Prewitt, Roberts and Laplacian), Gaussian Markov Random Fields (GMRF), Local Linear Transform.*
CCOM [66]	20 000	Heterogeneous	*Classification accuracy* × *query size*	N/A
Unser [67]	(1) 3 072, (2) 768 and (3) 192	Brodatz	*Classification accuracy*	GCOM
GCOM [32]	(1) 243, (2) 170, (3) 624	(1) Photomicrographs of sandstones, (2) Aerial photos, (3) Satellite images of California coastline	*Classification accuracy*	N/A
Takala LBP [68]	(1) 1 350, (2) 900	(1) Corel, (2) German stamps	Precision, Recall	*Color Correlogram*, EHD
LBP [34]	(1) 7 744, (2) 3 840 and 4 320	(1) Brodatz, (2) OUTex	*Classification accuracy*	Itself, *Rotation invariant wavelet*
IFLT [69]	(1) 672, (2) 3 840	(1) Brodatz; (2) OUTEX_TC_00010	*Classification precision*	Itself, LBP
SBPH [70]	640	VisTex	*Top-N average retrieval rate, with N=[16..64]*	*Original Steerable Pyramid Decomposition, GGD&KLD*
Han Gabor [39]	(1) 1 792, (2) 52	(1) Brodatz, (2) MPEG-7	Precision, Recall	Gabor
HTDI [38]	(1) 91, (2) 1 000	(1) Brodatz, (2) UIUCTex	Precision	HTD
M-SPyd [71]	(1) 208, (2) 208, (3) 208	(1) Brodatz, (2) Brodatz rotated, (3) Brodatz scaled	P_N, Showed 4 query results	*Original Steerable Pyramid Decomposition, Gabor Wavelets*
CSG [72]	2 400	Brodatz	*nT = precision or recall*	*Gradient vectors, Wavelet Energy Signature*
TBDI [57]	(1) 896, (2) 1 896	(1) Brodatz, (2) Corel	*Classification rate*, Extraction time (seconds), Showed query results	HTD
BSpline [73]	1 792	Brodatz	*Top-15 average retrieval rate*	Gabor, *DWT*
Kokare CW [74]	(1) 1 728, (2) 112, (3) 16	(1) Brodatz, (2) USC, (3) 1 artificial texture	*Top-16 average retrieval rate*	*Daubechies wavelets, Gabor wavelets*
Sim Zernike [75]	(1) 1 856, (2) 832, (3) 880, (4) 115, (5) 2 400, (6) 8 000, (7) 4 900	(1) Brodatz+USC, (2) ICU, (3) Rotated (Brodatz+ICU), (4) Scaled (Brodatz), (5) Corel, (6) LANSAT, (7) Rotated/Scaled (Brodatz+Corel)	Recall, Vector size (bytes), Extraction time (seconds), Distance complexity (number of sums and minus operations), *Entropy*	Gabor, *Radon, Wavelet*
MERF [76]	1 744	Brodatz	*Top-15 average retrieval rate*	Gabor, *MRSAR*
Sim DCT [77]	more than 3 000	Brodatz, ICU	Recall, Vector size (bits), Extraction time (seconds), Distance complexity (number of sums and minus operations)	Gabor, *Radon, Wavelet*
HTDR [37]	880	MPEG-7 (Brodatz=30+ICU=25)	*Top-15 average retrieval rate*, Number of distance computations, Distance time (seconds)	HTD original distance function, complete function
TBD [26, 35]	448	Brodatz	5 user subjective evaluations	N/A

...continues on next page...

...Table 4: continued from previous page...

Descriptor	**Collection size**	**Collection**[4]	**Evaluation measures**[5]	**Compared with**[5]
HTD [35, 28]	1 856	Brodatz	*Top-15 average retrieval rate*, Vector size (number of vector elements), Extraction time (seconds), Search and sort times (seconds)	*PWT, TWT, MRSAR*
Rubner EMD [78]	(1) 1 792 + others, (2) 500	(1) Brodatz, (2) Corel	Showed query results	N/A

4. Experimental comparison in small datasets

Given the large amount of descriptors selected for theoretical evaluation, we have to choose the most promising descriptors for the experimental comparison. In order to contrast with the large-scale Web-like environment, experiments were performed aiming to evaluate the descriptors' performance in a controlled small-scale scenario.

All those experiments were performed in the Eva tool[8], which creates an standardized environment for the evaluation of image descriptors. Details of Eva tool can be found in [79].

4.1. Selected descriptors

In this section we present the descriptors selected for the experimental evaluation, justifying each choice. The implementation details are shown in Sections 4.3 and 4.4 for color and texture descriptors, respectively.

4.1.1. Color descriptors

Global color histogram descriptor (GCH) [21], cumulative global color histogram (CGCH) [22], local color histogram descriptor (LCH) [21], and color coherence vectors (CCV) [56] are popular descriptors from literature, usually used as baseline for comparisons. LCH is a traditional descriptor based on fixed-size regions and CCV is segmentation-based. GCH, CGCH, and LCH are all very simple descriptors.

Descriptors based on correlograms are promising because they encode spatial information, as the traditional color autocorrelogram descriptor (ACC) [51] and the more elaborated joint autocorrelogram descriptor

[8]www.ic.unicamp.br/~penatti/eva/ (as of September 15, 2011.)

(JAC) [48]. Spatial information is usually lost when using simple color histograms.

The border/interior pixel classification descriptor (BIC) [55] and color-based clustering descriptor (CBC) [55] are descriptors based on segmentation. CBC has a complex extraction algorithm while BIC has a very simple one. BIC has also presented promising results in heterogeneous collections [55].

The color bitmap descriptor [52] was selected because it analyzes image color properties globally and by using fixed-size regions. Color structure descriptor (CSD) [26] was selected because, according to previous comparisons among the MPEG-7 color descriptors [13, 14], it achieves the best performance.

CW-HSV [47] and CW-LUV [46] descriptors are very simple and generate very compact feature vectors. The chromaticity moments descriptor (CM) [50] is very simple and generates compact feature vectors.

4.1.2. Texture descriptors

The local binary pattern descriptor (LBP) [34] is very popular in the literature and has simple algorithms presenting also good effectiveness and invariance to rotations and gray-scale variation. The homogeneous texture descriptor (HTD) [35, 28] is a traditional descriptor from MPEG-7 standard.

The Steerable Pyramid Decomposition descriptor with scale and rotation invariance (M-SPyd) [71] is used in our experimental comparison due to its good results [71] and its invariance to scale and rotation. The statistical analysis of structural information descriptor (SASI) [62] was chosen because it presented better results than descriptors based on Gabor filters, as reported in [62].

The color co-occurrence matrix (CCOM) [66] was chosen due to the popularity of the co-occurrence matrix for analyzing textures. It also aggregates color information to the original gray-level co-occurrence matrix (GCOM) [32]. Unser descriptor [67] was chosen due to its ability to generate more compact feature vectors, to have lower complexity and to present effectiveness similar to the GCOM descriptor.

The quantized compound change histogram (QCCH) [60] was chosen due to its simple extraction algorithm, its fast distance function and its compact feature vector. The main reason for choosing local activity spectrum descriptor (LAS) [65] was its simplicity for both feature vectors extraction and distance computations.

4.2. Evaluation measures

Efficiency in features extraction and distance computations. The times required for features extraction and distance computations of each descriptor were measured in *seconds*, *milliseconds* or *microseconds*. Time measures were taken by the Eva tool [79], which measures the average and standard deviation value for each feature vector extraction and for each distance computation. To minimize fluctuations on time measurements, Eva performs each feature vector extraction and each distance computation three times, keeping the average. From the absolute time values computed, an analysis considering the relative performance of the descriptors was conducted. Therefore, besides showing the absolute times of each descriptor, we highlight how much faster or slower each descriptor was in relation to a reference descriptor.

To take into account fluctuations in measured execution times, a very basic analysis was performed. A difference was considerable if it was bigger than the double of the maximum standard deviation for the two averages being compared: $|\mu_i - \mu_j| > 2 \times max(\sigma_i, \sigma_j)$, where μ and σ are respectively the average and standard deviation of the time measures.

Time measures were taken in a Dual 2.0 GHz Intel Xeon QuadCore computer with 4 GB of RAM, 3 hard disks in RAID-0 by hardware and Linux operating system. The implementations did not take advantage of the computer parallelism facilities.

Efficiency in feature vectors storage. The measures for the feature vectors sizes are *bytes* or *bits* multiples. Based on the number of values stored in the feature vectors, the feature vector size was computed. We considered that each value was a float or integer of 4 bytes each. The number of values in each feature vector was based on the quantization levels suggested in the original versions of the descriptors and used in our implementations.

Effectiveness. The effectiveness measures computed in our experiments were: Precision $\times$ Recall, mAP, P_1, P_5, P_{10}, R_1, R_5, and R_{10}. Precision $\times$ Recall and mAP, for the small-scale experiments, and P_1, P_5, P_{10}, R_1, R_5, and R_{10}, for the experiments in the Web scenario. The detailed definition of metrics and criteria is explained on Section 3.1.

Considering the Web-like scenario, it is important to emphasize that, as the dataset grows, only a small fraction is actually shown to the users in query results. Therefore, it is critical that the relevant images are ranked in

Table 5: Color descriptors implementation details.

Descriptor	**Color space**	**Quantization**	**Feature vector size (number of values)**	**Distance function**
ACC [51]	RGB	64 color, 4 distance (1, 3, 5, and 7)	256	L1
BIC [55]	RGB	64	128	dLog
CBC [24]	CIELab	N/A	variable (6 values per region)	L2 and Integrated Region Matching (IRM)
CCV [56]	RGB	64	128	L1
CGCH [22]	RGB	64	64	L1
CM [50]	CIE XYZ	N/A (color), 6 moments	12	L1
Color Bitmap [52]	RGB	N/A (color), 100 blocks	300 bits + 6 values	L2 and Hamming
CSD [26]	HMMD	184 (color), 8×8 neighborhood	184	L1
CW-HSV [47]	HSV	64	63 bits	Hamming
CW-LUV [46]	CIELuv	128	127 bits	Hamming
GCH [21]	RGB	64	64	L1
JAC [48]	HSV	64 (color), 5 (gradient magnitude, rank, and texturedness), 4 distance (1, 3, 5, and 7), neighborhood 5x5 (rank and texturedness)	32000	L1
LCH [21]	RGB	64 color, grid 4×4	1024	L1

the topmost positions. Considering a Precision × Recall curve, this desired behavior occurs when a descriptor presents high Precision values mainly for small Recall values.

4.3. Results for color descriptors

The 13 color descriptors presented in Section 4.1.1 were implemented according to the details showed in Table 5. For GCH and LCH we use L1 as the distance function.

The color descriptors were tested on ETH database (cropped-close). ETH is a free image database available on-line[9] that contains 3,280 full color images categorized in 8 classes (apples, pears, tomatoes, cows, dogs, horses, cups, and cars), each class containing 410 images. Each image is composed by one object in the center over a homogeneous background. The objects appear in different positions and point of views.

[9] http://www.d2.mpi-inf.mpg.de/Datasets/ETH80 (as of September 15, 2011.)

Table 6: Color descriptors comparison: absolute and relative times for (a) each feature vector extraction (in milliseconds), (b) for each distance computation (in microseconds), and (c) absolute and relative size of a single feature vector. Average values are shown with their respective standard deviation. Relative measures against GCH descriptor. Tables are sorted by the *average* column.

Descriptor	**Average**	**Rel.**
CGCH	0.3 ± 0.9	0.81
GCH	0.4 ± 1.1	1.00
LCH	0.6 ± 0.9	1.45
ColorBitmap	1.1 ± 1.3	2.69
BIC	2.3 ± 3.0	5.58
CW-HSV	2.4 ± 0.9	6.05
CM	2.6 ± 1.3	6.31
CCV	3.3 ± 0.9	8.18
CSD	15.0 ± 4.3	37.13
ACC	16.6 ± 8.8	40.89
CW-LUV	18.7 ± 1.4	46.21
JAC	31.4 ± 4.1	77.58
CBC	51.0 ± 8.2	126.01

(a) extraction times (milliseconds)

Descriptor	**Average**	**Rel.**
CM	28.6 ± 7.5	0.65
CW-HSV	29.9 ± 1.4	0.68
CW-LUV	32.0 ± 1.4	0.73
ColorBitmap	36.4 ± 2.9	0.83
CGCH	43.2 ± 2.0	0.98
GCH	43.8 ± 1.7	1.00
BIC	51.4 ± 1.6	1.17
CCV	52.1 ± 1.8	1.19
CSD	60.7 ± 2.0	1.38
ACC	71.8 ± 7.1	1.64
LCH	179.4 ± 17.7	4.09
JAC	788.8 ± 140.0	17.99
CBC	1154.7 ± 1470.9	26.34

(b) distance times (microseconds)

Descriptor	**Bytes**	**Rel.**
CW-HSV	7.875	0.03
CW-LUV	15.875	0.06
CM	48	0.19
ColorBitmap	61.5	0.24
GCH	256	1.00
CGCH	256	1.00
BIC	512	2.00
CCV	512	2.00
CSD	736	2.88
CBC	864	3.38
ACC	1024	4.00
LCH	4096	16.00
JAC	128000	500.00

(c) feature vector sizes

In the following sections, the results of the experimental evaluation with the color descriptors are presented. The relative values are computed using global color histogram (GCH) as the reference descriptor, because GCH is one of most popular descriptors from the literature and it is also often used as baseline for comparisons.

4.3.1. Efficiency

In features extraction. Table 6(a) shows the average time for extracting one feature vector, as well as the standard deviation, for each descriptor, sorted by average time. Table 6(a) also shows the relative times of each descriptor in relation to GCH descriptor. The values were obtained from a total of $9,840$ features extractions for each descriptor (3 times for each of the $3,280$ images).

Only CCV, CSD, CW-LUV, JAC, and CBC are significantly slower than GCH for features extraction. None of the tested color descriptors is significantly faster than GCH.

As observed, CBC is the most time-consuming descriptor for features extraction. In the theoretical evaluation, CBC is the most complex descriptor for features extraction, having complexity $O(n \log n)$, while all the other 12 descriptors are $O(n)$. Although all other descriptors are $O(n)$, there still were large differences in their actual time measurements.

In distance computations. Table 6(b) shows the average time for one distance computation and the standard deviation for each descriptor, sorted by average time. The times relative to GCH are also shown. The values were obtained from $32,275,200$ distance computations for each descriptor $(3,280 \times 3,280 \times 3)$.

The descriptors BIC, CCV, CSD, ACC, LCH, and JAC are considerably slower than GCH descriptor. On the other hand, CM, CW-HSV, CW-LUV, and Color Bitmap are considerably faster than GCH descriptor for distance computations.

According to the theoretical analysis, CBC was the only descriptor with distance function more complex than $O(vs)$. That higher complexity is observed in the times showed in Table 6(b) where CBC was the slowest descriptor. The differences in the times considering the other descriptors that are $O(vs)$ were basically due to the feature vector sizes.

In feature vectors storage. Considering the number of values in the feature vectors generated by each descriptor, we computed the feature vector sizes in bytes. Table 6(c) shows the absolute and relative feature vectors sizes of each color descriptor. As CBC descriptor generates variable-size feature vectors, the size showed in Table 6(c) is the average size for ETH database.

The descriptors ACC and JAC, that are based on the idea of correlogram, generate large feature vectors. JAC, in particular, generates very large feature vectors because it computes the joint autocorrelogram for more than one image property. The LCH descriptor also generates large feature vectors because it computes a color histogram for each image region.

To illustrate the impact of the differences in size of feature vectors in a Web scenario, assume a database with 10^9 images. CW-HSV descriptor would use nearly 7 GB of space for storing feature vectors while Color Bitmap would require 57 GB, GCH 238 GB, CSD 685 GB, and JAC would require 116 TB, for example.

4.3.2. Effectiveness

Figure 4 shows the average Precision $\times$ Recall curves for all the evaluated color descriptors considering all ETH images as queries.

For Recall value equal to 10% the best Precision values are very similar for BIC, CSD, JAC, ACC, and Color Bitmap descriptors. They are the only descriptors with Precision values over 70%. They remain the best descriptors until 50% of Recall. However, after 30% of Recall the Precision values are

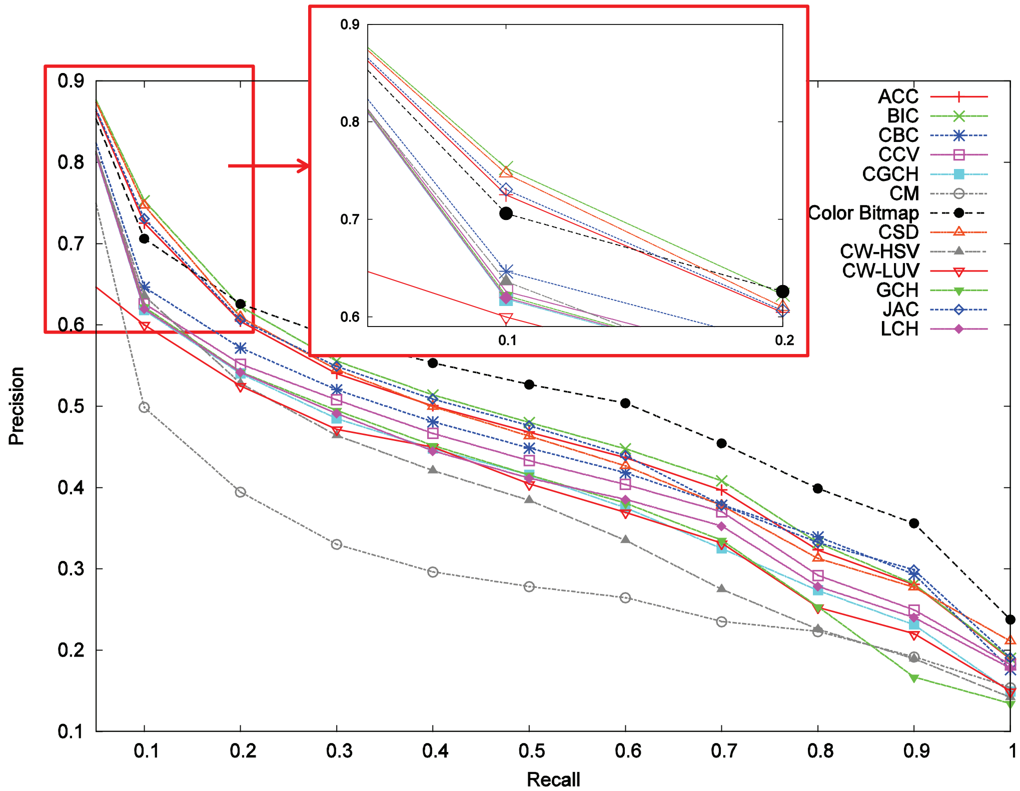


Figure 4: Precision × Recall curves for color descriptors in ETH database. The zoomed region represents the high-precision/low-recall zone, which is most important for a Web environment. (The graph is best viewed in color.)

similar for all descriptors (except for Color Bitmap, which is far above, and for CM, which is far below) which makes it difficult to point out which descriptors are the best.

Nevertheless, if we analyze the curves considering a Web environment, it is better if the descriptor has high Precision values when Recall values are small. The reason is because that only a small portion of the relevant images on the Web will be retrieved and showed to the user, which means small Recall. Consequently, we can consider BIC, CSD, JAC, ACC, and Color Bitmap as the most effective.

Table 7: Texture descriptors implementation details.

Descriptor	Parameters	Feature vector size (number of values)	Distance function
CCOM [66]	RGB quantized in 216 bins, d=1	variable	L1 similar
HTD [35, 28]	4 scales, 6 orientations	48	L1
LAS [65]	4 bins for each measure (non-uniformly quantized)	256	L1
LBP [34]	R=1, P=8	10	L1
M-SPyd [71]	2 scales, 4 orientations	16	L1
QCCH [60]	T=40	40	L1
SASI [62, 63]	windows: 3×3, 5×5, 7×7; 4 directions (0°, 45°, 90°, and 135°); therefore, K=8	64	dot distance
Unser [67]	Gray levels=256 bins, 4 directions (0°, 45°, 90°, and 135°), d=1.5; measures: mean, contrast, correlation, energy, entropy, homogeneity, maximum probability, and standard deviation.	32	L1

4.4. Results for texture descriptors

The 8 texture descriptors showed in Section 4.1.2 were implemented according to the details showed in Table 7.

Given our application scenario, we had to adjust some descriptors implementations. Our implementation of LBP uses values 1 for the parameter R (radio) and 8 for P (neighbors). Also, we use L1 distance instead of the original proposed for LBP, which is non-symmetric. The original M-SPyd descriptor, has invariances to scale and rotation independently. In the implemented version, however, both invariances are computed at the same time. In other words, the feature vector generated by our implementation of M-SPyd is invariant to scale and rotation at the same time. HTD, M-SPyd and SASI had database dependent normalization steps which were forgone, as they are incompatible with the highly dynamic Web-like scenario intended. Our implementation of LAS descriptor quantizes components non-uniformly, to avoid generating histograms with an overwhelming majority of values in the first bin. When the descriptor proposed no distance function (which was the case for Unser), we used L1.

The texture descriptors were tested on Brodatz database. Brodatz [58] is one of the most popular image databases for the evaluation of texture descriptors. As observed in Section 3.3, Brodatz collection was the most widely used in the evaluation of the texture descriptors present on this work. Brodatz contains 111 different textures. As done in the great majority of papers, each texture is divided into fixed-size cells. Our experiments divided

Table 8: Texture descriptors comparison: absolute and relative times for (a) each feature vector extraction (in milliseconds), (b) for each distance computation (in microseconds), and (c) absolute and relative size of a single feature vector. Average values are shown with their respective standard deviation. Relative measures against LBP descriptor. Tables are sorted by the *average* column.

Descriptor	Average	Rel.
LBP	1.7 ± 0.4	1.00
CCOM	3.1 ± 0.6	1.84
Unser	3.3 ± 0.5	2.02
LAS	5.1 ± 1.5	3.09
QCCH	21.8 ± 1.0	13.15
M-SPyd	218.2 ± 0.7	131.54
SASI	430.3 ± 0.9	259.39
HTD	4892.6 ± 49.6	2949.22

(a) extraction times (milliseconds)

Descriptor	Average	Rel.
Unser	31.3 ± 1.3	0.90
M-SPyd	31.9 ± 1.4	0.92
HTD	31.9 ± 1.5	0.92
QCCH	32.2 ± 1.6	0.93
LAS	32.8 ± 1.3	0.95
SASI	33.9 ± 1.0	0.98
LBP	34.6 ± 1.3	1.00
CCOM	42.2 ± 1.9	1.22

(b) distance times (microseconds)

Descriptor	Bytes	Rel.
LBP	40	1.00
M-SPyd	64	1.60
Unser	128	3.20
QCCH	160	4.00
HTD	192	4.80
CCOM	248	6.20
SASI	256	6.40
LAS	1024	25.60

(c) feature vector sizes

each texture in 16 blocks, totalizing $1,776$ images. The resulting database is composed by 111 categories having 16 images each.

In the following sections, we present the evaluation results of texture descriptors. The relative measures consider LBP descriptor as reference, due to its current popularity and its simplicity.

4.4.1. Efficiency

In features extraction. Table 8(a) shows the average times for extracting one feature vector and the standard deviation values for each descriptor, ordered by average time. The values were computed from $5,328$ features extractions (3 times for each of the $1,776$ images). Relative times are also shown.

All descriptors are significantly slower than LBP for features extraction. As observed, the descriptors M-SPyd, SASI, and HTD are considerably slower than the others. As presented in the theoretical comparison in Section 3.3, those three descriptors have higher complexity than the other five descriptors compared here. That high complexity resulted in their larger times for features extraction. HTD is very time-consuming when extracting feature vectors, being yet more than 10 times slower than SASI.

The five descriptors with complexity $O(n)$ presented little variation among themselves.

In distance computations. The average time and standard deviation for one distance computation by each descriptor is shown in Table 8(b), ordered by average time. The times relative to LBP are also shown. The values were obtained from $9,462,528$ distance computations $(1,776 \times 1,776 \times 3)$.

The difference in times relative to LBP is considerable only for Unser and CCOM descriptors. Unser is faster than LBP and CCOM is slower.

Although there exist some difference in the times measured for each descriptor, it is small. The theoretical analysis showed that those 8 descriptors have distance function with linear complexity. That fact is observed in the times measured, as they are similar among the descriptors. The small differences are related to the feature vector sizes.

In feature vectors storage. The feature vector sizes of each descriptor are presented in Table 8(c). The sizes were computed based on the parameters given in Section 4.1.2 as explained in Section 4.2. CCOM descriptor generates variable-size feature vectors, thus the size showed in Table 8(c) is the average size for Brodatz database.

The biggest difference in feature vectors sizes is for LAS descriptor. Although it generates feature vectors more than 25 times larger than LBP's feature vectors, the size is equal to ACC color descriptor, which is a feasible size in many situations.

The differences in feature vectors sizes become evident when we consider a Web environment. Considering a database with 10^9 images, LBP descriptor would require nearly 37 GB for storing feature vectors, while HTD would use 179 GB, SASI 238 GB, and LAS 954 GB, for example.

4.4.2. Effectiveness

For the effectiveness evaluation, all images from Brodatz collection were used as query images. Figure 5 shows the average values among all queries performed.

LAS and SASI descriptors are clearly superior in terms of effectiveness than the other descriptors. Considering small Recall values, the three best descriptors are LAS, SASI, and CCOM.

5. Evaluation in a Web-like environment

All the 13 color and 8 texture descriptors were evaluated in a Web-like environment. That environment is mainly characterized by a huge image database with highly heterogeneous content and with no "fixed" or "final" ground-truth giving an authoritative classification of the images.

The database used was collected by researchers from Federal University of Amazonas (UFAM), Brazil, with the objective to create a collection with representative data from the Web. The data gathering started recursively from

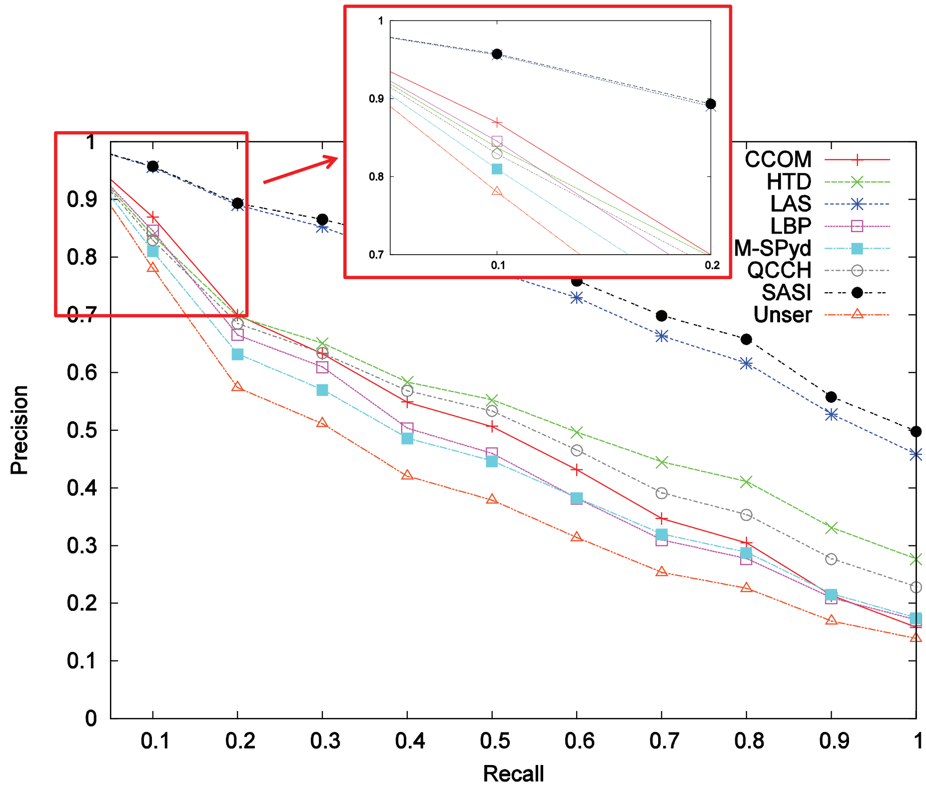


Figure 5: Precision × Recall for texture descriptors in Brodatz database. The zoomed region represents the high-precision/low-recall zone, which is most important for a Web environment. (The graph is best viewed in color.)

the Yahoo directory and generated a database with 234,143 images (excluding icons and banners) and 1.2 million HTML documents. The database contains all kinds of images, but no semantical labeling. From now on, we refer to this database as *WebSample*.

As we did for the previous experiments, efficiency and effectiveness measures were computed for each descriptor. The reference descriptor is global color histogram (GCH).

5.1. Efficiency

For the efficiency measures, a subset of 500 images was randomly selected from the original WebSample database.

Table 9: Descriptors comparison in the Web environment: absolute and relative times for (a) each feature vector extraction (in milliseconds), (b) for each distance computation (in microseconds), and (c) absolute and relative size of a single feature vector. Average values are shown with their respective standard deviation. Relative measures against GCH descriptor. Tables are sorted by the *average* column. Texture descriptors appear in italic.

Descriptor	Average	Rel.
CGCH	5.7 ± 29.7	0.8
GCH	7.4 ± 32.8	1.0
ColorBitmap	8.7 ± 33.4	1.2
LCH	11.1 ± 37.7	1.5
LBP	19.8 ± 53.7	2.7
BIC	20.8 ± 51.6	2.8
CCOM	23.4 ± 53.6	3.1
Unser	39.9 ± 78.8	5.4
CW-HSV	40.3 ± 80.4	5.4
CM	44.6 ± 85.8	6.0
LAS	52.7 ± 96.2	7.1
CCV	62.3 ± 108.2	8.4
CSD	86.1 ± 72.6	11.6
ACC	87.3 ± 147.1	11.7
QCCH	239.1 ± 359.3	32.2
CW-LUV	285.9 ± 435.1	38.4
JAC	487.8 ± 737.4	65.6
CBC	817.5 ± 1340.1	109.9
M-SPyd	2289.0 ± 3867.9	307.8
SASI	5101.3 ± 7222.1	686.1
HTD	19811.2 ± 25819.6	2664.4

(a) extraction times (milliseconds)

Descriptor	Average	Rel.
CM	34.0 ± 1.2	0.67
CW-LUV	34.7 ± 1.3	0.69
CW-HSV	35.5 ± 1.2	0.70
Unser	36.8 ± 1.2	0.73
QCCH	37.3 ± 1.3	0.74
M-SPyd	37.6 ± 1.2	0.74
SASI	38.0 ± 1.2	0.75
HTD	38.0 ± 1.2	0.75
LAS	38.9 ± 1.3	0.77
CCOM	41.7 ± 2.8	0.82
LBP	41.8 ± 1.2	0.83
ColorBitmap	42.8 ± 1.4	0.85
GCH	50.5 ± 1.3	1.00
CGCH	52.1 ± 1.6	1.03
BIC	61.8 ± 1.3	1.22
CCV	64.0 ± 1.7	1.27
CSD	74.1 ± 1.8	1.47
ACC	87.1 ± 1.4	1.72
LCH	230.1 ± 1.9	4.55
JAC	2023.1 ± 625.0	40.03
CBC	11856.6 ± 10281.1	234.61

(b) distance times (microseconds)

Descriptor	Bytes	Rel.
CW-HSV	7.88	0.03
CW-LUV	15.88	0.06
LBP	40	0.16
CM	48	0.19
ColorBitmap	61.5	0.24
M-SPyd	64	0.25
Unser	128	0.50
QCCH	160	0.63
HTD	192	0.75
CGCH	256	1.00
GCH	256	1.00
SASI	256	1.00
BIC	512	2.00
CCV	512	2.00
CSD	736	2.88
CCOM	999.61	3.90
ACC	1024	4.00
LAS	1024	4.00
LCH	4096	16.00
CBC	6456.47	25.22
JAC	128000	500.00

(c) feature vector sizes

In features extraction. Table 9(a) presents the average time for extracting one feature vector by each descriptor, as well as the standard deviation and the relative times. The values were computed from 1,500 features extractions for each descriptor (3 times for each of the 500 images).

WebSample has, in general, larger images and more image size variability than the dedicated databases used earlier. That is reflected in the larger averages and standard deviations reported in the extraction times.

No descriptor was considerably faster or slower than the others, although they present large differences in the average times. For example, the texture descriptors M-SPyd, SASI, and HTD are very slow to process one image. HTD specially takes almost 20 seconds, on average, to extract one feature vector. CBC descriptor is also slow, taking almost 1 second to process one image, on average. Considering a Web search engine, M-SPyd, SASI, and HTD face considerable challenges in order to be employed.

In distance computations. Table 9(b) shows the average times and standard deviation values for one distance computation, ordered by average time. The values were obtained from $750,000$ distance computations ($500 \times 500 \times 3$) for each descriptor.

The times observed were similar to the small-scale experiments on the previous section because most of the descriptors generate fixed-size feature vectors. CBC descriptor, however, generates variable-size feature vectors and their distance times were more than 10 times larger in WebSample than in ETH database. To compare one pair of feature vectors, CBC takes, on average, more than 10 seconds, making its use on large-scale environments very challenging. JAC descriptor is also challenging for a Web environment, as it takes more than 2 seconds, on average, to compare each pair of feature vectors.

There are considerable differences in distance times between the descriptors in relation to GCH. All the descriptors above GCH in Table 9(b) are faster and all below are slower, except for CGCH and CBC.

In feature vectors storage. The feature vectors sizes shown in Table 9(c) were computed with the same parameters used for the small-scale experiments. The sizes are the same for all descriptors, except for CBC and CCOM. These two descriptors, as discussed, generate variable-size feature vectors (whose average sizes for WebSample are shown in Table 9(c)) and reveals a large increase in comparison to the sizes in ETH and Brodatz databases, respectively. CBC vectors increased, on average, more than 7 times and CCOM vectors, more than 4 times. CBC vectors are larger because its clustering algorithm groups similar color and WebSample database has many more complex images than ETH database. CCOM vectors are larger because color is present in WebSample images, while it was absent in Brodatz textures.

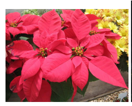
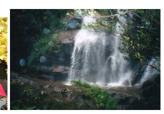
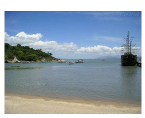
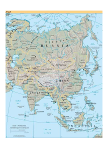
Table 9(c) shows that there is little correlation between vector size and the type of descriptor (color or texture).

JAC descriptor generates by far the largest feature vectors. That could have an important impact in the storage requirements of a CBIR system.

5.2. Effectiveness

The effectiveness evaluation in WebSample collection was based on a set of queries with an associated pool of relevants for each query [80]. This set has 30 images and each query has a pool of relevant images that were selected by real users. The query images are shown in Table 10. The main reason for

Table 10: Query images, their identification numbers, and the sizes of their respective pool of relevants.

						
number	1	2	3	4	5	6
Pool size	15	8	14	19	12	39
						
number	7	8	9	10	11	12
Pool size	15	11	50	9	18	17
						
number	13	14	15	16	17	18
Pool size	14	22	52	21	19	29
						
number	19	20	21	22	23	24
Pool size	16	37	24	34	27	17
						
number	25	26	27	28	29	30
Pool size	73	33	25	10	42	11

this kind of evaluation is that WebSample database has no standard categorization. Moreover, involving users in the evaluation process can measure descriptor effectiveness in a real scenario, taking into account the semantic variability of final users judgment.

To create the pool of relevants, the user-oriented evaluation interface provided by Eva tool [79] was used. A set of human evaluators composed by 18 people (graduate and undergraduate students) analyzed the images retrieved by a set of descriptors [80] and marked them as relevant or non-relevant for each query. It is important to say that there was no information about which descriptor retrieved each image and no extra information was given about

Table 11: Average P_1, P_5, P_{10}, R_1, R_5, and R_{10} values over all the query images for WebSample database, sorted by P_{10}. Texture descriptors appear in italic.

Descriptor	$\mathbf{P_1}$	$\mathbf{P_5}$	$\mathbf{P_{10}}$	$\mathbf{R_1}$	$\mathbf{R_5}$	$\mathbf{R_{10}}$
JAC	0.667	0.433	**0.287**	0.036	0.118	0.146
BIC	0.567	0.367	**0.287**	0.029	0.085	0.141
ACC	0.500	0.367	**0.267**	0.025	0.092	0.138
LCH	0.467	0.320	**0.200**	0.028	0.092	0.109
CCOM	0.367	0.227	**0.167**	0.017	0.055	0.077
CCV	0.333	0.180	**0.113**	0.013	0.042	0.053
CSD	0.233	0.153	**0.103**	0.012	0.041	0.054
HTD	0.233	0.147	**0.093**	0.011	0.031	0.038
GCH	0.367	0.113	**0.090**	0.020	0.031	0.049
SASI	0.233	0.120	**0.080**	0.011	0.034	0.040
CW-HSV	0.167	0.107	**0.070**	0.010	0.023	0.030
LAS	0.200	0.073	**0.053**	0.010	0.021	0.028
CBC	0.100	0.073	**0.043**	0.007	0.024	0.030
LBP	0.133	0.067	**0.040**	0.005	0.015	0.016
Color Bitmap	0.067	0.053	**0.033**	0.003	0.012	0.014
CW-LUV	0.133	0.060	**0.033**	0.009	0.017	0.019
CGCH	0.100	0.033	**0.023**	0.005	0.007	0.010
QCCH	0.167	0.040	**0.023**	0.008	0.010	0.010
M-SPyd	0.067	0.027	**0.020**	0.004	0.006	0.009
Unser	0.033	0.013	**0.013**	0.002	0.002	0.005
CM	0.033	0.007	**0.003**	0.002	0.002	0.002

each query image. Therefore, the users had no influence when interpreting the query image. Table 10 shows the sizes of each pool of relevants.

Using the pool of relevant images for each query, the effectiveness evaluation could be conducted automatically by analyzing the images retrieved by each descriptor and verifying if they are present in the pool of the corresponding query. The pool sizes shown in Table 10 include the query image. However, for the computation of the effectiveness measures, the query was ignored from the pool.

Table 11 presents the average P_1, P_5, P_{10}, R_1, R_5, and R_{10} values among all query images for each descriptor, while Figure 6 shows the box plot for each descriptor considering the P_{10} for all 30 queries. In general, the measures tell that the effectiveness of the descriptors is poor. On average, the best descriptors are able to retrieve around 2 relevant images between the 10 first results. However, it is important to note that the descriptors evaluated here

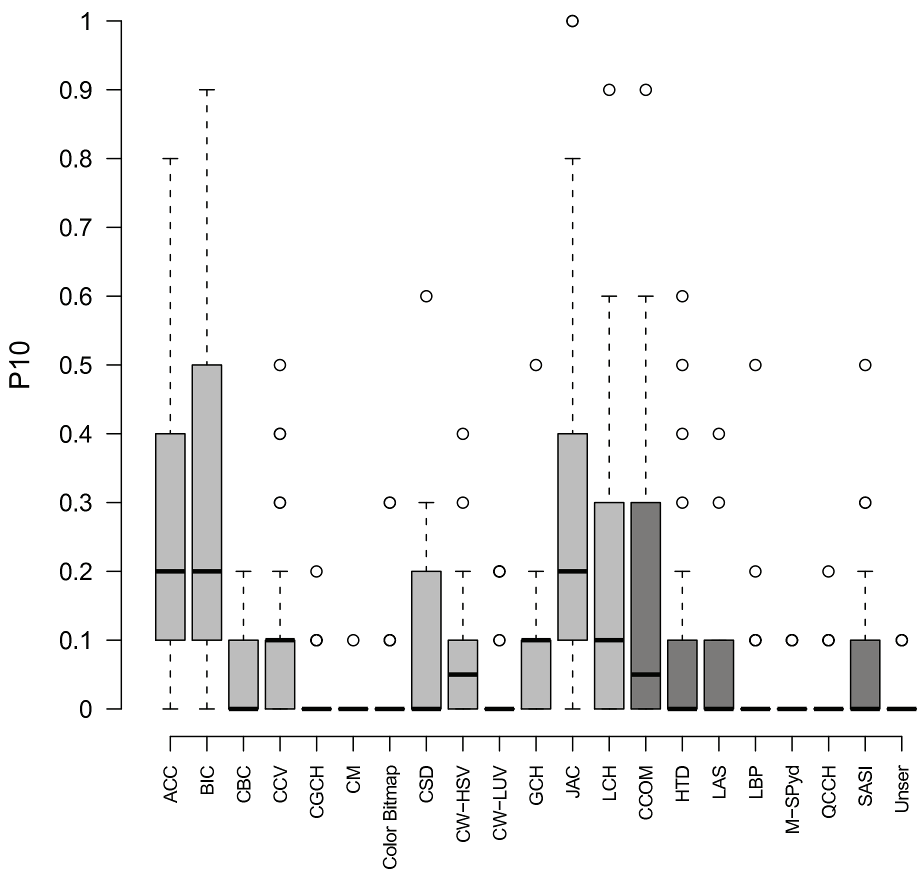


Figure 6: Box plot with each descriptor P_{10} values on all 30 queries. Values above or below 1.5 the interquartile range were considered outliers and indicated as small circles. Color descriptors appear in light gray while texture ones appear in the right in dark gray. Descriptors are sorted alphabetically.

characterize the image information globally, without having any notion of which image region is more important for the user.

The precision and recall values indicate that the best overall effectiveness is achieved by descriptors JAC, BIC, and ACC, which have reached more than 26% of P_{10} and more than 13% of R_{10}, in the average case. The three

Table 12: Average P_{10} and R_{10} values for queries 11 and 29, sorted by P_{10}. Texture descriptors appear in italic.

query 11

Descriptor	$\mathbf{P_{10}}$	$\mathbf{R_{10}}$
LCH	0.90	0.53
JAC	0.80	0.47
ACC	0.70	0.41
CSD	0.60	0.35
BIC	0.50	0.29
HTD	0.50	0.29
LAS	0.40	0.24
CCOM	0.30	0.18
CCV	0.30	0.18
SASI	0.30	0.18
GCH	0.20	0.12
CW-HSV	0.10	0.06
CBC	0.00	0.00

(a)

query 29

Descriptor	$\mathbf{P_{10}}$	$\mathbf{R_{10}}$
JAC	1.00	0.24
BIC	0.90	0.22
CCOM	0.90	0.22
ACC	0.70	0.17
HTD	0.60	0.15
LCH	0.60	0.15
CW-HSV	0.40	0.10
CCV	0.30	0.07
CSD	0.20	0.05
LAS	0.10	0.02
CBC	0.00	0.00
GCH	0.00	0.00
SASI	0.00	0.00

(b)

are all color descriptors. The best texture descriptor is CCOM with average P_{10} over 16%. Local color histogram (LCH) also has average P_{10} of 20%.

Figure 6 shows that many descriptors present very poor effectiveness, having P_{10} values close to zero for most of the queries. This was the case for descriptors CGCH, CM, Color Bitmap, CW-LUV, LBP, M-SPyd, QCCH, and Unser. We can observe that there is a large variation in the behavior of most descriptors, having queries with both very low and very high precisions.

ACC, BIC, and JAC, which present the best average precisions, also have similar medians in Figure 6, except that BIC has more queries with higher P_{10} values. For example, JAC reached P_{10} of 100% for one query, while BIC, LCH, and CCOM reached P_{10} of 90% in one case.

To illustrate the effectiveness of the descriptors in some queries, we have selected four queries, corresponding to the two easiest queries, and two cases that illustrate well the problem of semantic gap. The worst descriptors in terms of effectiveness (CGCH, CM, Color Bitmap, CW-LUV, LBP, M-SPyd, QCCH, and Unser) were eliminated from the following analysis since their precision was zero or close to zero for almost all queries considered.

Table 12 shows the precision and recall values for the two easiest queries. Despite that, only 6 descriptors had a top-10 precision above 50%, showing

Table 13: Average P_{10} and R_{10} values for queries 5 and 15, sorted by P_{10}. Texture descriptors appear in italic.

query 5

Descriptor	$\mathbf{P}_{10}$	$\mathbf{R}_{10}$
BIC	0.20	0.18
JAC	0.20	0.18
ACC	0.00	0.00
CW-HSV	0.10	0.09
CCOM	0.10	0.09
SASI	0.10	0.09
CBC	0.00	0.00
CCV	0.00	0.00
CSD	0.00	0.00
GCH	0.00	0.00
LCH	0.00	0.00
HTD	0.00	0.00
LAS	0.00	0.00

(a)

query 15

Descriptor	$\mathbf{P}_{10}$	$\mathbf{R}_{10}$
ACC	0.70	0.14
BIC	0.60	0.12
CCV	0.40	0.08
JAC	0.30	0.06
LCH	0.30	0.06
CCOM	0.20	0.04
CSD	0.10	0.02
CW-HSV	0.10	0.02
GCH	0.10	0.02
HTD	0.10	0.02
CBC	0.00	0.00
LAS	0.00	0.00
SASI	0.00	0.00

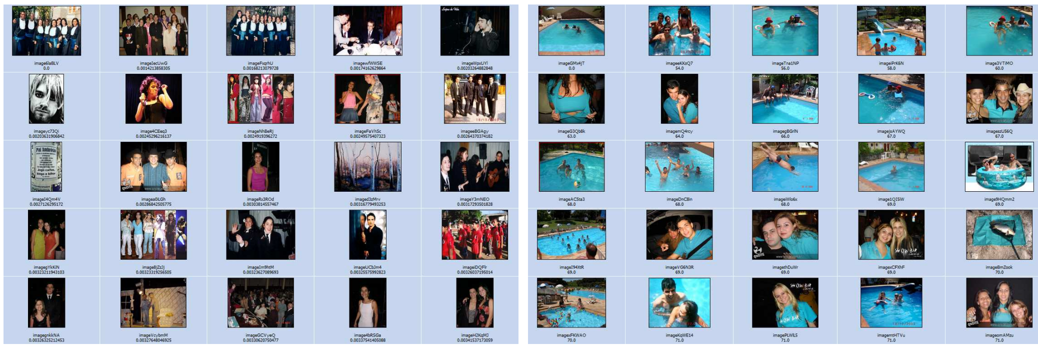
(b)

the challenge of the task. Query 11 is very stereotypical, and query 29 has a large pool of relevant answers, which are both factors that help to ease the task. However, we have observed that those are neither necessary nor sufficient conditions to explain query difficulty.

Table 13 and Figure 7 shows the results for two queries that illustrate well the semantic gap phenomenon[10].

In general, color descriptors performed better than texture descriptors, but there was a large variability among queries. As we hinted above, the most common explanations (size of the pool of relevants, simplicity of the image) were not sufficient to explain that variability. We have observed that the specific choice of the query image may have an important effect on the user interpretation of its contents, which poses a challenge for Query-by-Example systems based on a single query, and reinforces the importance of query refinement/user feedback.

[10]We have provided an interface for the visualization of all queries results: http://www.recod.ic.unicamp.br/eva/yahoo230k (as of September 15, 2011.)



(a) (b)

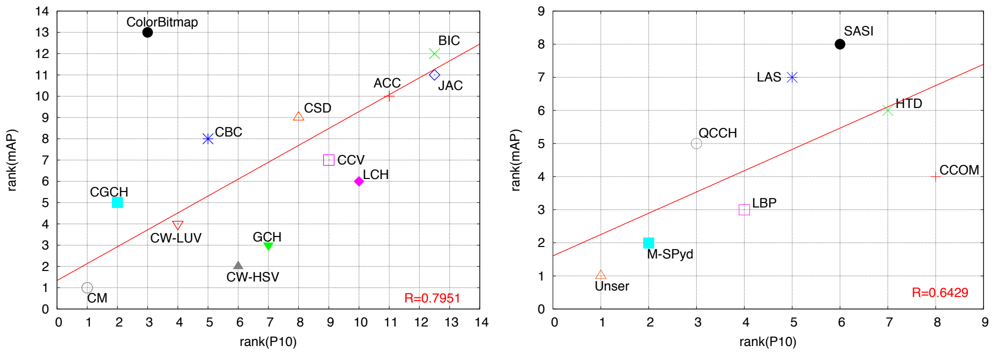
Figure 7: Results for query 5 and 15 showing the semantic gap phenomenon. The query image is the first in the top-left position of the grid. In (a), there are many images having people, but only one of them with people graduating. In (b), there are some images very similar in color properties, but not necessarily in semantics. Images retrieved by SASI and BIC, respectively.

5.3. Comparing effectiveness in small and large-scale experiments

In general, it is remarkable the effect of heterogeneity in the descriptors' effectiveness. To evaluate that effect, we have computed the correlation between the effectiveness measures in the small-scale experiments and the results in the WebSample database. Figure 8 shows the correlation between the rank of the descriptors in each scenario and plots the linear regression model for them. The rank is based on the mAP for the small-scale and the P_{10} for the large-scale experiment. For both color and texture descriptors, the correlation of relative effectiveness on the two experiments is significant, but with notable departs.

Color Bitmap relative effectiveness is so much better in the small-scale experiment than in the large-scale one, that we had to consider it an outlier in the regression computation (with it, the correlation coefficient falls to $R = 0.5199$). Other notable departs are the LCH, GCH, CW-HSV, which perform relatively better in the large-scale experiment.

For texture descriptors, the correlation observed is smaller, reflecting the more specific nature both of many texture representation and the special-purpose Brodatz dataset. CCOM was one of few texture descriptors designed with heterogeneous database in mind. Accordingly, it has the best relative



(a) (b)

Figure 8: Correlation of effectiveness results of (a)color and (b)texture descriptors between small-scale experiments and the experiments in the Web scenario. The correlation is based on the rank of the descriptors in each experiment. The correlation coefficient and the regression for the data are also shown. For the color descriptors regression, Color Bitmap was considered an outlier. Note that the performance in small and large-scale experiments has significant correlation, but there are important deviations for both color and texture descriptors.

performance on the large-scale experiment, though its performance on the Brodatz dataset is unremarkable.

5.4. Descriptors combination

Considering the challenges of retrieving relevant images in a Web scenario, the combination of more than one descriptor is often recommended. The exact way descriptors should be combined (e.g., early fusion of the feature vectors, late fusion of the distances, adaptive methods, etc.) in order to achieve maximum performance is an open research question and outside the scope of this paper.

However, we follow the example of [3] and provide an indicative metric, considering the degree of agreement among the best descriptors studied over the queries, which is useful to determine when the combination can be useful. Descriptors with a high degree of agreement tend to reinforce each other, while descriptors with a low agreement can be used to complement each other and both effects must be taken into consideration when designing combination algorithms.

Table 14: Correlation coefficients between each pair of the five best descriptors considering the P_{10} values over the queries in WebSample database. Texture descriptors appear in italic.

	ACC	BIC	JAC	LCH	*CCOM*
ACC	-	-	-	-	-
BIC	0.801	-	-	-	-
JAC	0.578	0.670	-	-	-
LCH	0.661	0.478	0.610	-	-
CCOM	0.659	0.712	0.622	0.431	-

To avoid disrupting the analysis, we have chosen only the best-performing descriptors, and computed their correlations over the queries on the WebSample dataset. That pairwise analysis is shown in Table 14.

A more delicate analysis take into account all descriptors at once and studies the cross-correlations using a Principal Component Analysis. The two main components (which account for 82% of the variability) are plotted on Figure 9, and show that two groups of descriptors tend to agree: JAC and LCH, on one hand, and ACC, BIC and CCOM on the other hand. Note also that the overall agreement between those best descriptors is significant.

The plot in Figure 9 also reveal how individual queries performed on those descriptors. Horizontally, queries to the left of the plot tended to be easier than queries to the right. Vertically, queries to the center tended to perform equally well (or bad) for all descriptors, while queries on top or bottom showed the largest differences.

5.5. Conclusion

The experiments in the Web environment have shown interesting aspects of the descriptors performances. Considering efficiency, it was noticed an increase in extraction times mainly due to the larger dimensions of WebSample images. Some descriptors like M-SPyd, SASI, HTD, and CBC face challenges in a Web search engine, because of their large extraction times.

Considering the times to compute distances, changing the scenario caused very few impact for most of the descriptors, because most of them generate fixed-size feature vectors. For the descriptors whose vectors sizes depend on the complexity of the input image, some of them have suffered an increase in distance times more than others. Descriptors like CBC and JAC show times challenging for a Web scenario.

The effectiveness results have shown a real impact when changing the scenario from specific to broad. Overall, descriptors performance fall consid-

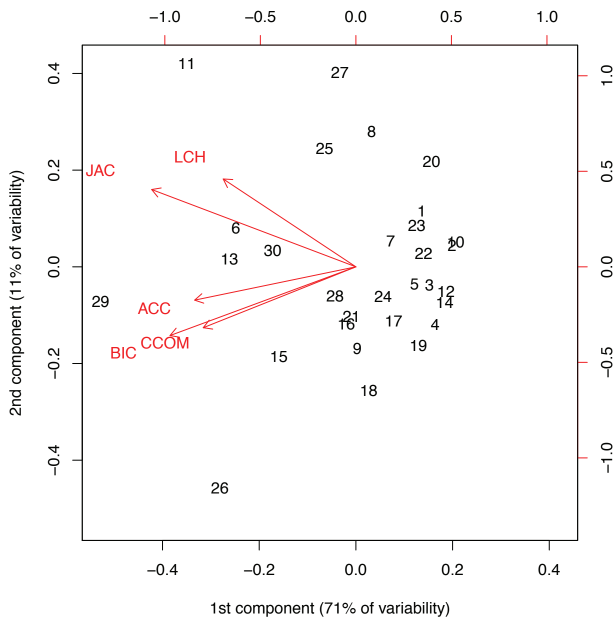


Figure 9: Plot of the two main component of a Principal Component Analysis of the P_{10} values over the queries in the WebSample database. Those two components explain 82% of the variability and indicate the two groups of descriptors that tend to perform similarly: JAC and LCH versus ACC, BIC and CCOM. The small numbers identify the individual queries.

erably, making many of them not adequate for the Web environment. Such decrease may come from both the size and heterogeneity of the database, which aggravate the semantic gap in images, since their interpretation by users in an open setting is more subtle than in a specific context.

We have noticed the influence of the image background in the descriptors' effectiveness. Heterogeneous backgrounds, with colors and textures that mix up with the image main object, affected the values generated in the feature vectors. While the human vision easily discerns the background and the main object, the extraction algorithms usually do not distinguish what is in evidence in an image. In query images with homogeneous background, the feature vectors generated were more representative for the image's main object.

Considering all the experimental results, we could point 5 good descriptors: JAC, BIC, ACC, LCH, and CCOM. However, considering the Web scenario, JAC would face the challenge of the efficiency problems presented. LCH and ACC are slower than BIC for distance computations while ACC and CCOM generate larger feature vectors than BIC. LCH generates larger vectors than BIC, CCOM, and ACC. BIC presented better effectiveness. Therefore, BIC is the most recommended to be used in a Web environment, as it presented better measures in almost all aspects.

Among the reasons for BIC good performance is its extraction algorithm. The classification of the pixels in interior or border is similar to a simple segmentation between homogeneous regions and textured regions. Consequently, some texture information is also encoded into the feature vector. In other cases, this simple segmentation can characterize the image objects separately from the image background, having a histogram for the background, usually composed by interior pixels, and a histogram for the objects, usually composed by border pixels.

A simple analysis considering the combination potential was performed with the best descriptors revealing collective correlations that should be taken into account for combination techniques.

6. Discussion and conclusions

The existence of a huge amount of visual information available on the Web nowadays demands the creation of efficient and effective retrieval systems. Content-based image retrieval (CBIR) systems are suitable for this task. The main aspect of CBIR systems is the ability to search and index images by their visual properties, like color, texture, shape and spatial relationship of objects. In a CBIR system, the image descriptor has a fundamental role, as it is responsible for measuring images visual similarities.

This paper presents a comparative study of global color and texture descriptors considering the Web as environment of use. Web presents an extremely large and heterogeneous scenario. This paper also serve as reference for developers and researchers looking for image descriptors suitable for heterogeneous systems.

The comparative study considers theoretical and experimental aspects, using efficiency and effectiveness criteria. The study also shows taxonomies for color and texture description. The theoretical comparison is conducted over 24 color and 28 texture descriptors, including traditional and recently proposed descriptors. The theoretical evaluation shows that most of the descriptors have $O(n)$ complexity for features extraction and have also linear complexity for distance computations. The most promising descriptors in the theoretical analysis were implemented and evaluated experimentally. In the experiments it was possible to measure the efficiency in features extraction, in distance computations, in storing feature vectors, and the effectiveness of the descriptors. Relating theoretical and experimental evaluation, we have noticed that descriptors with high theoretical complexity presented poor efficiency performance in the experiments.

Experiments were conducted over specific and heterogeneous collections. After evaluating color and texture descriptors separately in specific environments, the descriptors were all evaluated in a Web scenario, by using a collection with more the 230 thousand images with very heterogeneous content. Efficiency and effectiveness measures were computed for each descriptor. The effectiveness evaluation considered a pool of queries with their relevant sets created by real users. The correlation between results in specific and Web scenarios is not neglectable, but show important deviations, that should be taken into account when selecting the descriptors.

Color descriptors have presented precision values over 70% in ETH database, while in Web sample collection the average precision was under 30%. Texture descriptors have shown precision values above 80% in Brodatz collection, however, in the Web scenario the average precision was under 20%. In general, color descriptors suffer less effectiveness degradation in relation to the texture descriptors.

The experimental results in the Web scenario show that BIC [55] and ACC [51] are the best choices for an heterogeneous environment. BIC presents good efficiency results keeping one of the highest effectiveness. ACC is slower than BIC for features extraction and distance computations and generates larger feature vectors. The *color co-occurrence matrix* (CCOM) [66] would be the next choice. CCOM is faster than ACC for features extraction and faster than BIC for distance computations, however it generates larger feature vectors than BIC and its effectiveness is not as good as the two others.

Overall, the semantic gap continues to be a big challenge for image descriptors, especially in the context of information retrieval (in opposition to image classification with large learning sets). Users assignment of answer relevance based on a single image is subtle, because obviously, a single image may represent many different concepts. The necessity to represent a main object capturing the concept of interest over variable complex backgrounds is also a challenge.

We expect that relevance feedback techniques, query expansion and query refinement, computer vision-based approaches, and methodologies able to effectively combine different descriptors will continue to be important in order to improve retrieval effectiveness.

7. Acknowledgments

We would like to thank Fapesp (grant numbers 2006/59525-1, 2009/10554-8, and 2009/05951-8) CNPq, Capes, and Microsoft Research for infrastructure and financial support. We thank the colleagues who have helped with the descriptors implementations. We also thank the researchers from Federal University of Amazons (UFAM), Federal University of Minas Gerais (UFMG) and University of Campinas (Unicamp) who have given us the pool of queries to be used in our experiments.

References

[1] M. L. Kherfi, D. Ziou, A. Bernardi, Image retrieval from the world wide web: Issues, techniques, and systems, ACM Computing Surveys 36 (1) (2004) 35–67.

[2] R. Datta, D. Joshi, J. Li, J. Z. Wang, Image retrieval: Ideas, influences, and trends of the new age, ACM Computing Surveys 40 (2008) 5:1–5:60.

[3] T. Deselaers, D. Keysers, H. Ney, Features for image retrieval: an experimental comparison, Information Retrieval 11 (2) (2008) 77–107.

[4] S. Loncaric, A Survey of Shape Analysis Techniques, Pattern Recognition 31 (8) (1998) 983–1190.

[5] D. Zhang, G. Lu, Review of Shape Representation and Description, Pattern Recognition 37 (1) (2004) 1–19.

[6] M. Safar, C. Shahabi, X. Sun, Image retrieval by shape: A comparative study, in: IEEE International Conference on Multimedia and Expo, 2000, pp. 141–144.

[7] D. Zhang, G. Lu, A comparative study of fourier descriptors for shape representation and retrieval, in: Asian Conference on Computer Vision, 2002, pp. 646–651.

[8] D. Zhang, G. Lu, Evaluation of mpeg-7 shape descriptors against other shape descriptors, Multimedia Systems 9 (1) (2003) 15–30.

[9] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recognition 29 (1) (1996) 51–59.

[10] F. Xu, Y. Zhang, Evaluation and comparison of texture descriptors proposed in mpeg-7, Journal of Visual Communication and Image Representation 17 (4) (2006) 701–716.

[11] P. Howarth, S. M. Rüger, Evaluation of texture features for content-based image retrieval, in: International Conference on Image and Video Retrieval, 2004, pp. 326–334.

[12] O. A. B. Penatti, R. d. S. Torres, Color descriptors for web image retrieval: A comparative study, in: Brazilian Symposium on Computer Graphics and Image Processing, 2008, pp. 163–170.

[13] J. Annesley, J. Orwell, J.-P. Renno, Evaluation of mpeg7 color descriptors for visual surveillance retrieval, in: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005, pp. 105–112.

[14] T. Ojala, M. Aittola, E. Matinmikko, Empirical evaluation of mpeg-7 xm color descriptors in content-based retrieval of semantic image categories, in: International Conference on Pattern Recognition, Vol. 2, 2002, pp. 1021–1024.

[15] T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors: a survey, Foundations and Trends in Computer Graphics and Vision 3 (2008) 177–280.

[16] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 1615–1630.

[17] K. E. A. van de Sande, T. Gevers, C. G. M. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 1582–1596.

[18] Y. Wang, F. Makedon, R-histogram: quantitative representation of spatial relations for similarity-based image retrieval, in: ACM International Conference on Multimedia, 2003, pp. 323–326.

[19] R. da S. Torres, A. X. Falcão, Content-Based Image Retrieval: Theory and Applications, Revista de Informática Teórica e Aplicada 13 (2) (2006) 161–185.

[20] Y. Rubner, C. Tomasi, L. J. Guibas, The earth mover's distance as a metric for image retrieval, International Journal of Computer Vision 40 (2000) 99–121.

[21] M. J. Swain, D. H. Ballard, Color indexing, International Journal of Computer Vision 7 (1) (1991) 11–32.

[22] M. A. Stricker, M. Orengo, Similarity of color images, in: SPIE Storage and Retrieval for Image and Video Databases III, Vol. 2420, 1995, pp. 381–392.

[23] R. de O. Stehling, M. A. Nascimento, A. X. Falcão, Cell histograms versus color histograms for image representation and retrieval, Knowledge and Information Systems 5 (3) (2003) 315–336.

[24] R. de O. Stehling, M. A. Nascimento, A. X. Falcao, An adaptive and efficient clustering-based approach for content-based image retrieval in image databases, in: International Database Engineering & Applications Symposium, 2001, pp. 356–365.

[25] Y. Deng, B. S. Manjunath, C. Kenney, M. S. Moore, H. Shin, An efficient color representation for image retrieval, IEEE Transactions on Image Processing 10 (1) (2001) 140–147.

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

[27] M. Tuceryan, A. K. Jain, Texture analysis (1993) 235–276.

[28] B. S. Manjunath, W. Y. Ma, Texture features for browsing and retrieval of image data, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (8) (1996) 837–842.

[29] J. A. M. Zegarra, N. J. Leite, R. da S. Torres, Wavelet-based Feature Extraction for Fingerprint Image Retrieval, Journal of Computational and Applied Mathematics 227 (2009) 294–307.

[30] A. del Bimbo, Visual Information Retrieval, Morgan Kaufmann Publishers, 1999.

[31] H. Tamura, S. Mori, T. Yamawaki, Textural features corresponding to visual perception, IEEE Transactions on Systems, Man and Cybernetics 8 (6) (1978) 460–473.

[32] R. M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, IEEE Transactions on Systems, Man and Cybernetics 3 (6) (1973) 610–621.

[33] R. Nevatia, Machine Perception, Prentice-Hall, 1982.

[34] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 971–987.

[35] P. Wu, B. S. Manjunath, S. Newsam, H. D. Shin, A texture descriptor for browsing and similarity retrieval, Signal Processing: Image Communication 16 (1-2) (2000) 33–43.

[36] U. A. Ahmad, K. Kidiyo, R. Joseph, Texture features based on fourier transform and gabor filters: an empirical comparison, in: International Conference on Machine Vision, 2007, pp. 67–72.

[37] Y. M. Ro, H. K. Kang, Hierarchical rotational invariant similarity measurement for mpeg-7 homogeneous texture descriptor, Electronics Letters 36 (15) (2000) 1268–1270.

[38] L. Zhang, J. Ma, X. Xu, B. Yuan, Rotation invariant image classification based on mpeg-7 homogeneous texture descriptor, in: International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Vol. 3, 2007, pp. 798–803.

[39] J. Han, K. Ma, Rotation-invariant and scale-invariant gabor features for texture image retrieval, Image and Vision Computing 25 (9) (2007) 1474–1481.

[40] S. Barton, V. Gouet-Brunet, M. Rukoz, C. Charbuillet, G. Peeters, Estimating the indexability of multimedia descriptors for similarity searching, in: Adaptivity, Personalization and Fusion of Heterogeneous Information, 2010, pp. 84–87.

[41] H. Lejsek, F. H. Ásmundsson, B. T. Jónsson, L. Amsaleg, Scalability of local image descriptors: a comparative study, in: International Conference on Multimedia, 2006, pp. 589–598.

[42] K. Wong, L. Po, K. Cheung, A compact and efficient color descriptor for image retrieval, in: IEEE International Conference on Multimedia and Expo, 2007, pp. 611–614.

[43] N. Yang, W. Chang, C. Kuo, T. Li, A fast mpeg-7 dominant color extraction with new similarity measure for image retrieval, Journal of Visual Communication and Image Representation 19 (2) (2008) 92–105.

[44] H. A. Moghaddam, T. T. Khajoie, A. H. Rouhi, M. S. Tarzjan, Wavelet correlogram: A new approach for image indexing and retrieval, Pattern Recognition 38 (12) (2005) 2506–2518.

[45] X. Li, Image retrieval based on perceptive weighted color blocks, Pattern Recognition Letters 24 (12) (2003) 1935–1941.

[46] K. Nallaperumal, M. S. Banu, C. C. Christiyana, Content based image indexing and retrieval using color descriptor in wavelet domain, in: International Conference on Computational Intelligence and Multimedia Applications, Vol. 3, 2007, pp. 185–189.

[47] A. Utenpattanant, O. Chitsobhuk, A. Khawne, Color descriptor for image retrieval in wavelet domain, in: International Conference on Advanced Communication Technology, Vol. 1, 2006, pp. 818–821.

[48] A. Williams, P. Yoon, Content-based image retrieval using joint correlograms, Multimedia Tools and Applications 34 (2) (2007) 239–248.

[49] T. Chaira, A. K. Ray, Fuzzy measures for color image retrieval, Fuzzy Sets and Systems 150 (3) (2005) 545–560.

[50] G. Paschos, I. Radev, N. Prabakar, Image content-based retrieval using chromaticity moments, IEEE Transactions on Knowledge and Data Engineering 15 (5) (2003) 1069–1072.

[51] J. Huang, S. R. Kumar, M. Mitra, W. Zhu, R. Zabih, Image indexing using color correlograms, in: International Conference on Computer Vision and Pattern Recognition, 1997, pp. 762–768.

[52] T. Lu, C. Chang, Color image retrieval technique based on color features and image bitmap, Information Processing and Management 43 (2) (2007) 461–472.

[53] J. Sun, X. Zhang, J. Cui, L. Zhou, Image retrieval based on color distribution entropy, Pattern Recognition Letters 27 (10) (2006) 1122–1126.

[54] H. Y. Lee, H. K. Lee, Y. H. Ha, Spatial color descriptor for image retrieval and video segmentation, IEEE Transactions on Multimedia 5 (3) (2003) 358–367.

[55] R. de O. Stehling, M. A. Nascimento, A. X. Falcão, A compact and efficient image retrieval approach based on border/interior pixel classification, in: International Conference on Information and Knowledge Management, 2002, pp. 102–109.

[56] G. Pass, R. Zabih, J. Miller, Comparing images using color coherence vectors, in: ACM International Conference on Multimedia, 1996, pp. 65–73.

[57] K. Lee, L. Chen, An efficient computation method for the texture browsing descriptor of mpeg-7, Image and Vision Computing 23 (5) (2005) 479–489.

[58] P. Brodatz, Textures: A Photographic Album for Artists and Designers, Dover, 1966.

[59] S. Kiranyaz, M. Ferreira, M. Gabbouj, A generic shape/texture descriptor over multiscale edge field: 2-d walking ant histogram, IEEE Transactions on Image Processing 17 (3) (2008) 377–391.

[60] C. Huang, Q. Liu, An orientation independent texture descriptor for image retrieval, in: International Conference on Communications, Circuits and Systems, 2007, pp. 772–776.

[61] E. Hadjidemetriou, M. D. Grossberg, S. K. Nayar, Multiresolution histograms and their use for recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (7) (2004) 831–847.

[62] A. Çarkacioglu, F. Yarman-Vural, Sasi: a generic texture descriptor for image retrieval, Pattern Recognition 36 (11) (2003) 2615–2633.

[63] A. Çarkacioglu, F. Yarman-Vural, Sasi: a new texture descriptor for content based image retrieval, in: International Conference on Image Processing, Vol. 2, 2001, pp. 137–140.

[64] J. Zhou, L. Xin, D. Zhang, Scale-orientation histogram for texture image retrieval, Pattern Recognition 36 (4) (2003) 1061–1063.

[65] B. Tao, B. W. Dickinson, Texture recognition and image retrieval using gradient indexing, Journal of Visual Communication and Image Representation 11 (3) (2000) 327–342.

[66] V. Kovalev, S. Volmer, Color co-occurence descriptors for querying-by-example, MultiMedia Modeling 0 (1998) 32–38.

[67] M. Unser, Sum and difference histograms for texture classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (1) (1986) 118–125.

[68] V. Takala, T. Ahonen, M. Pietikäinen, Block-based methods for image retrieval using local binary patterns, in: Scandinavian Conference on Image Analysis, 2005, pp. 882–891.

[69] P. Janney, Z. Yu, Invariant features of local textures - a rotation invariant local texture descriptor, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–7.

[70] J. A. M. Zegarra, J. Beeck, N. J. Leite, R. da S. Torres, A. X. Falcão, Combining global with local texture information for image retrieval applications, in: International Symposium on Multimedia, Vol. 0, 2008, pp. 148–153.

[71] J. A. M. Zegarra, N. J. Leite, R. d. S. Torres, Rotation-invariant and scale-invariant steerable pyramid decomposition for texture image retrieval, in: Brazilian Symposium on Computer Graphics and Image Processing, 2007, pp. 121–128.

[72] P. W. Huang, S. K. Dai, P. L. Lin, Texture image retrieval and image segmentation using composite sub-band gradient vectors, Journal of Visual Communication and Image Representation 17 (5) (2006) 947–957.

[73] Q. Wang, D. D. Feng, Z. Chi, B-spline over-complete wavelet based fractal signature analysis for texture image retrieval, in: International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004, pp. 462–466.

[74] M. Kokare, B. N. Chatterji, P. K. Biswas, Cosine-modulated wavelet based texture features for content-based image retrieval, Pattern Recognition Letters 25 (4) (2004) 391–398.

[75] D. Sim, H. Kim, R. Park, Invariant texture retrieval using modified zernike moments, Image and Vision Computing 22 (4) (2004) 331–342.

[76] X. Yang, J. Liu, Maximum entropy random fields for texture analysis, Pattern Recognition Letters 23 (1-3) (2002) 93–101.

[77] D. Sim, H. Kim, R. Park, Fast texture description and retrieval of dct-based compressed images, Electronics Letters 37 (1) (2001) 18–19.

[78] Y. Rubner, C. Tomasi, Texture-based image retrieval without segmentation, in: IEEE International Conference on Computer Vision, Vol. 2, 1999, p. 1018.

[79] O. A. B. Penatti, R. d. S. Torres, Eva - an evaluation tool for comparing descriptors in content-based image retrieval tasks, in: ACM International Conference on Multimedia Information Retrieval, 2010, pp. 413–416.

[80] P. A. S. Kimura, J. M. B. Cavalcanti, P. C. Saraiva, R. da S. Torres, M. A. Gonçalves, Evaluating retrieval effectiveness of descriptors for searching in large image databases, Journal of Information and Data Management (to appear).