Content-Based Image Retrieval: Theory and Applications

Ricardo da Silva Torres [1]
Alexandre Xavier Falcão [1]

Abstract:
Advances in data storage and image acquisition technologies have enabled the creation of large image datasets. In this scenario, it is necessary to develop appropriate information systems to efficiently manage these collections. The commonest approaches use the so-called *Content-Based Image Retrieval (CBIR) systems*. Basically, these systems try to retrieve images similar to a user-defined specification or pattern (e.g., shape sketch, image example). Their goal is to support image retrieval based on *content* properties (e.g., shape, color, texture), usually encoded into *feature vectors*. One of the main advantages of the CBIR approach is the possibility of an automatic retrieval process, instead of the traditional keyword-based approach, which usually requires very laborious and time-consuming previous annotation of database images. The CBIR technology has been used in several applications such as fingerprint identification, biodiversity information systems, digital libraries, crime prevention, medicine, historical research, among others.
This paper aims to introduce the problems and challenges concerned with the creation of CBIR systems, to describe the existing solutions and applications, and to present the state of the art of the existing research in this area.

Keywords: content-based image retrieval, image database, image descriptors, indexing, query specification, query visualization, effectiveness measures, relevance feedback, applications.

1 Introduction

Advances in data storage and image acquisition technologies have enabled the creation of large image datasets. In order to deal with these data, it is necessary to develop appropriate information systems to efficiently manage these collections. Image searching is one of the most important services that need to be supported by such systems. In general, two different approaches have been applied to allow searching on image collections: one based on image textual medatada and another based on image content information.

The first retrieval approach is based on attaching textual metadata to each image and uses traditional database query techniques to retrieve them by keywords [1, 2]. However,

[1] Institute of Computing, State University of Campinas, Campinas, SP, Brazil.
{rtorres,afalcao}@ic.unicamp.br

these systems require a previous annotation of the database images, which is a very laborious and time-consuming task. Furthermore, the annotation process is usually inefficient because users, generally, do not make the annotation in a systematic way. In fact, different users tend to use different words to describe a same image characteristic. The lack of systematization in the annotation process decreases the performance of the keyword-based image search.

These shortcomings have been addressed by the so-called *Content-Based Image Retrieval (CBIR) systems* [3–5]. In these systems, image processing algorithms (usually automatic) are used to extract feature vectors that represent image properties such as color, texture, and shape. In this approach, it is possible to retrieve images similar to one chosen by the user (*query-by-example*). One of the main advantages of this approach is the possibility of an automatic retrieval process, contrasting to the effort needed to annotate images.

The creation of CBIR systems (e.g., [3, 6]) involves research on databases and image processing, handling problems that vary from storage issues to friendly user interfaces [4]. Images are particularly complex to manage – besides the volume they occupy, retrieval is an application-and-context-dependent task [5]. It requires the translation of high-level user perceptions into low-level image features (this is the so-called "semantic gap" problem). Moreover, image indexing is not just an issue of string processing (which is the case of standard textual databases). To index visual features, it is common to use numerical values for the n features and then to represent the image or object as a point in a *n-dimensional* space [7]. Multi-dimensional indexing techniques [8, 9] and common similarity metrics [10] are factors to be taken into account. In this context, the main challenges faced are the specification of indexing structures to speed up image retrieval and the query specification as a whole. Furthermore, query processing also depends on cognitive aspects related to visual interpretation. Several other problems – query languages, data mining – contribute to attract computer scientists to this area.

This paper aims to introduce the problems and challenges concerned with the creation of CBIR systems, to describe the existing solutions and applications, and to present the state of the art of the existing research in this area.

This article introduces the basic concepts of the CBIR domain in Section 2. Section 3 presents an overview of the existing techniques for creating CBIR systems, while Section 4 discusses user interaction in CBIR systems. Some applications that take advantage of the CBIR technnolgy are discussed in Section 5. Next, open research problems in this area are presented in Section 6. Section 7 states our conclusions.

2 Basic Concepts

This section presents several constructs to handle image-content descriptors and related services. Firstly, Section 2.1 presents the typical architecture of CBIR systems. In the following, the main components of this architecture are formalized in Section 2.2.

2.1 Architecture of CBIR Systems

Figure 1 shows a typical architecture of a content-based image retrieval system. Two main functionalities are supported: data insertion and query processing.

The data insertion subsystem is responsible for extracting appropriate features from images and storing them into the image database (see dashed modules and arrows). This process is usually performed off-line.

The query processing, in turn, is organized as follows: the interface allows a user to specify a query by means of a query pattern and to visualize the retrieved similar images. The query-processing module extracts a feature vector from a query pattern and applies a metric (such as the Euclidean distance) to evaluate the similarity between the query image and the database images. Next, it ranks the database images in a decreasing order of similarity to the query image and forwards the most similar images to the interface module. Note that database images are often indexed according to their feature vectors by using structures such as M-tree [11] or Slim-tree [12] to speed up retrieval and similarity computation.

Note that both the data insertion and the query processing functionalities use the feature vector extraction module.

2.2 Formalization

2.2.1 Image, Feature Vectors, and Image Descriptors A typical *CBIR* solution requires the construction of an **image descriptor**, which is characterized by: (i) an *extraction algorithm* to encode image features into *feature vectors*; and (ii) a *similarity measure* to compare two images. The similarity measure is a *matching function*, which gives the degree of similarity for a given pair of images as represented by their feature vectors, often defined as an inverse function of the distance (e.g., Euclidean), that is, the larger the distance value, the less similar the images are.

More formally:

Streams are sequences of elements of an arbitrary type (e.g., bits, characters, images, etc.). A **stream** is a $sequence$ whose codomain is a nonempty set [13].

An **image *stream*** (or simply **image**) $\hat{I}$ is a pair $(D_I, \vec{I})$, where:

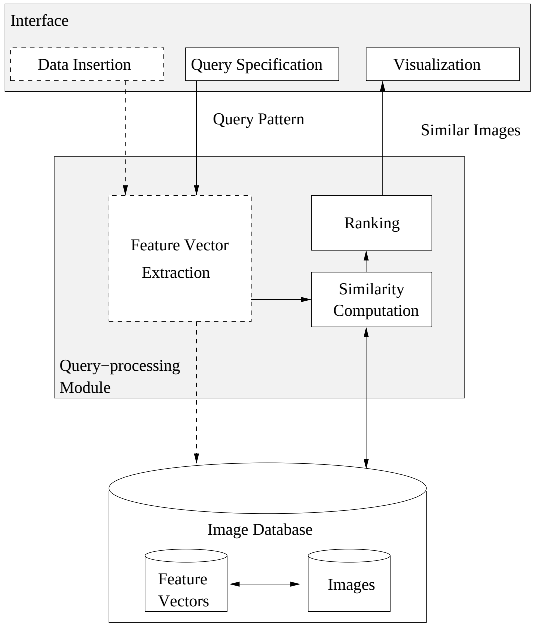


Figure 1. Typical architecture of a content-based image retrieval system.

- D_I is a finite set of *pixels* (points in $\mathbb{N}^2$, that is, $D_I \subset \mathbb{N}^2$), and
- $\vec{I} : D_I \rightarrow \mathbb{R}^n$ is a function that assigns to each pixel p in D_I a vector $\vec{I}(p) \in \mathbb{R}^n$ (for example, $\vec{I}(p) \in \mathbb{R}^3$ when a color in the RGB system is assigned to a pixel).

A **feature vector** $\vec{v}_{\hat{I}}$ of an image $\hat{I}$ can be thought of as a point in $\mathbb{R}^n$ space: $\vec{v}_{\hat{I}} = (v_1, v_2, ..., v_n)$, where n is the dimension of the vector.

Examples of possible feature vectors are a color histogram [14], a multiscale fractal curve [15], and a set of Fourier coefficients [16]. They essentially encode image properties, such as color, shape, and texture. Note that different types of feature vectors may require different similarity functions.

A **simple image content descriptor** (briefly, **image descriptor**) D is defined as a tuple (ϵ_D, δ_D), where:

- $\epsilon_D : \{\hat{I}\} \rightarrow \mathbb{R}^n$ is a function, which extracts a *feature vector* $\vec{v}_{\hat{I}}$ from an *image* $\hat{I}$.

- $\delta_D : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ is a *similarity function* (e.g., based on a distance metric) that computes the similarity between two images as the inverse of the distance between their corresponding *feature vectors*.

Figure 2 illustrates the use of a simple descriptor D to compute the similarity between two images $\hat{I}_A$ and $\hat{I}_B$. First, the extraction algorithm ϵ_D is used to compute the feature vectors $\vec{v}_{\hat{I}_A}$ and $\vec{v}_{\hat{I}_B}$ associated with the images. Next, the similarity function δ_D is used to determine the similarity value d between the images. Eventually, multiple descriptors can be combined into a complex descriptor, which is able to encode multiple image properties at a same time [17].

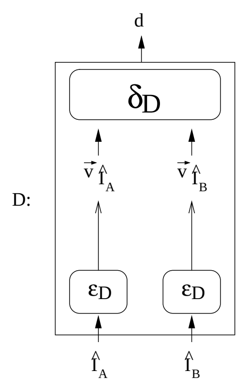


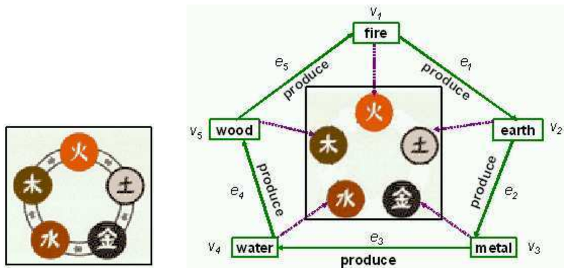
Figure 2. The use of a simple descriptor D for computing the similarity between images.

2.2.2 Query Specification and Result Visualization A conceptual representation for user information need is materialized into a query specification. A **query specification** Q is a tuple $Q = \{(H_q, Contents_q, P_q)\}$, where $H_q = ((V_q, E_q), L_q, F_q)$ is a structure (i.e., a directed graph with vertices V_q and edges E_q, along with labels L_q and labeling function F_q on the graph; see Def. 2 in [13] for details), $Contents_q$ includes digital objects and all of their streams, and P_q is a mapping function $P_q : V_q \rightarrow Contents_q$.

The notion of conceptual representation for user information needs was used in [13] to define a searching service, however, it was not formally defined. The formal definition for conceptual representations for user information needs was introduced in [18].

An example of a query specification is: $q = (H_q, Contents_q, P_q) \in Q$. For example: q is an image, which contains five spatially related sub-images (objects). A user wants to find some images similar to an existing one as shown in Fig. 3(a). Thus, $q = ((V_q, E_q), L_q, F_q), Contents_q, P_q)$, where $V_q = v_1, v_2, v_3, v_4, v_5$, $E_q = e_1, e_2, e_3, e_4, e_5$,

$L_q = 'fire', 'earth', 'metal', 'water', 'wood', 'produce'$, $F_q : V_q \cup E_q \rightarrow L_q$, $Contents_q$ is the stream of the five spatially related sub-images with their location information, and $P_q : V_q \rightarrow Contents_q$ (see Fig. 3(b)).



(a) (b)

Figure 3. (a) q is an image of (b) 5 spatially related sub-images (See [18] for more details).

Usually, two kinds of queries are supported by CBIR systems [11]. In a *K-nearest neighbor query (KNNQ)*, the user specifies the number k of images to be retrieved that are closest to the query pattern. In a *range query (RQ)*, the user defines a search radius r and wants to retrieve all database images whose distances to the query pattern are less than r. In this case, both the specification of k in the KNNQ and the specification of r needs to be incorporated into Q.

A query specification $q \in Q$ is a **K-nearest neighbor query (KNNQ) information need** if there exists $v \in V_q$, a real number $k \in Contents_q$, and $P_q(v) = k$.

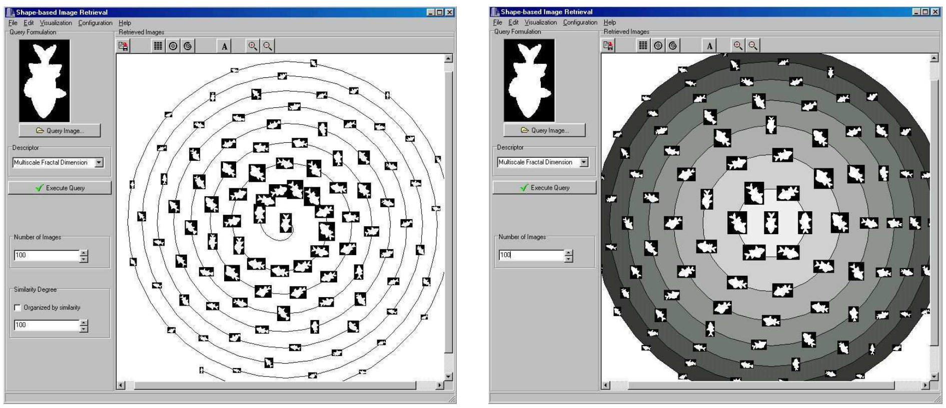
A query specification $q \in Q$ is a **range query (RQ) information need** if there exists $v \in V_q$, a real number $r \in Contents_q$, and $P_q(v) = r$.

Let C be a collection (a set of digital objects; see Def. 17 in [13] for details), and 2^C be the set of all subsets of C.

Let V_{spa} be a vector space (see Def. 13 in the appendix of [13]) and $Base$ be a set of basis vectors in V_{spa}. Let $\{VisualM\}$ be a set of visual marks (e.g., points, lines, areas, volumes, and glyphs) and $\{VisualMP\}$ be a set of visual properties (e.g., position, size, length, angle, slope, color, gray scale, texture, shape, animation, blink, and motion) of visual marks.

A **visualization operation** OP_{viz} is a set of functions $OP_{viz} = \{VisualMap_1, VisualMap_2, VisualMap_3\}$, where $VisualMap_1 : 2^C \rightarrow V_{spa}$ associates a set of digital objects with a set of vectors; $VisualMap_2 : 2^C \rightarrow VisualM$ associates a set of digital objects with a type of visual mark; $VisualMap_3 : Base \rightarrow VisualMP$ associates a basis vector with a visual property of a visual mark. For more details, see [18].

Fig. 4 shows two examples of the use of OP_{viz} to visualize results in a shape-based image retrieval system [19]. Each of the returned images is mapped to a vector in a vector space $Vspa$ by function $VisualMap_1$. $VisualMap_2$ maps returned images to thumbprints. In Figure 4(a), a function $VisualMap_3'$ is used to present the most similar images. This function places the query image in the center, and fills a spiral line with the retrieved images at regular distances, in a decreasing order of their similarity to the query image [19]. In Figure 4(b) a function $VisualMap_3''$ is used to present the most similar images in concentric rings. In this case, the rings are filled from the innermost ring to the outermost one, according to the image ranking [19].



(a) (b)

Figure 4. (a) Spiral approach. (b) Concentric rings approach.

2.2.3 Image Collection and Image Searching Service An **image collection** $ImgC$ is a tuple $(C, S_{imgdesc})$, where C is a collection (see Def. 17 in [13]) and $S_{imgdesc}$ is a set of image descriptors.

A **content-based image searching service** is a set of searching scenarios $\{sc_1$, sc_2, $\ldots$, $sc_t\}$, where each scenario sc_i is a sequence of events, and each event e_i is associated with the OP_s function defined as follows:

$OP_s : (Q \times C) \times Sim_s \rightarrow 2^{Contents}$, where $Sim_s = OP_q(q, ido)|q \in Q, ido \in C$, and where $OP_q : Q \times C \rightarrow R$ is a matching function that associates a real number with $q \in Q$ and a digital object $ido \in C$. The computation of OP_q relies on the use of appropriate image descriptors (e.g., their extraction and distance computation algorithms) defined in the image collection $ImgC$.

The range of function OP_s is the $Contents$ associated with collection $ImgC$. While the similarity function OP_q was defined in Def. 21 in [13], the retrieved results were not defined there. We consider the retrieved results as (a subset of) the $Contents$.

Note that OP_{viz} may be applied to the result of OP_s. In this sense, it can be seen as the final event of a content-based image searching service.

3 Related Work

This section aims to present a brief overview of existing approaches in the CBIR area.

3.1 Image Descriptors

An image descriptor is a pair, *feature vector extraction function* and *distance function*, used for image indexation by similarity. The extracted feature vector subsumes the image properties and the distance function measures the dissimilarity between two images with respect to their properties. This section aims to present a brief overview of existing image descriptors.

3.1.1 Color Descriptors: Color property is one of the most widely used visual feature in content-based image retrieval (CBIR) systems. Researches in this field can be grouped into three main subareas: (a) definition of adequate color space for a given target application, (b) proposal of appropriate extraction algorithms, and (c) study/evaluation of similarity measures.

Color information is represented as points in three-dimensional color spaces (such as RGB, HSV, YIQ, $L^*u^*v^*$, $L^*a^*b^*$ [20]). They allow discrimination between color stimuli and permit similarity judgment and identification [20]. Some of them are hardware-oriented (e.g., RGB, and CMY color space), as they were defined by taking into account properties of the devices used to reproduce colors. Others are user-inspired (e.g., $L^*u^*v^*$, $L^*a^*b^*$) as they were defined to quantify color differences as perceived by humans.

Several color description techniques have been proposed [14, 21–24]. They can be grouped into two classes based on whether or not they encode information related to the color spatial distribution.

Examples of descriptors that do not include spatial color distribution include Color Histogram and Color Moments. **Color Histogram** [14] is the most commonly used descriptor in image retrieval. The color histogram extraction algorithm can be divided into three steps: partition of the color space into cells, association of each cell to a histogram bin, and counting of the number of image pixels of each cell and storing this count in the corresponding histogram bin. This descriptor is invariant to translation and rotation. The similarity between two color histograms can be performed by computing the L_1, L_2, or weighted Euclidean distances, as well as by computing their intersection [14].

Other example of descriptor that does not consider color spatial distribution are the so-called **Color Moments** [21]. Usually, the *mean* (first order), *variance* (second), and *skewness* (third) are used to form the feature vector. These moments are defined, respectively, as $E_i = (1/N)\sum_{j=1}^{N} p_{ij}$, $\sigma_i = \sqrt[2]{(1/N)\sum_{j=1}^{N}(p_{ij} - E_i)^2}$, and $s_i = \sqrt[3]{(1/N)\sum_{j=1}^{N}(p_{ij} - E_i)^3}$, where p_{ij} is the value of the i-th color component of the image pixel j, and N is the number of the pixels in the image.

Examples of color descriptors that incorporate color spatial distribution include **Color Coherence Vector (CCV)** [23], **Border/Interior Pixel Classification (BIC)** [25], and **Color Correlogram** [22]. CCVs are created by computing, for each color, the total number of coherent (α_i) and incoherent pixels (β_i). A pixel is considered coherent if it belongs to a largely uniformly-colored region. The CCV is defined as $V_c =< (\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_N, \beta_N) >$, where N is the number of colors. The color correlogram, in turn, encodes the spatial correlation of colors. It can be seen as a table γ indexed by color pairs. Given any pixel of color c_i in the image, $\gamma^{(k)}_{c_i,c_j}$ gives the probability that a pixel at distance k away from the given pixel is of color c_j. The color correlogram is a table indexed by color pair, where the k-th entry for $< i, j >$ specifies the probability of finding a pixel of color j at a distance k from a pixel of color i in the image. In the BIC approach, each image pixel is classified as a border or interior pixel, based on whether it is at the border of the image itself or if at least one of its 4-neighbors have a different color. In the following, two histograms are computed: one considering only border pixels and another for only interior pixels.

The MPEG-7 initiative [26, 27], formally known as Multimedia Content Description Interface, focuses on the description of multimedia content, including content of various modalities like image, video, speech, graphics and their combinations. One of the most important components of the MPEG-7 framework is the proposal of image descriptors. For the color property, MPEG-7 has defined a number of histogram descriptors, a dominant color descriptor, and a color layout descriptor [24].

3.1.2 Texture Descriptors: There is no widely accepted definition of texture. However, this image property can be characterized by the existence of basic primitives, whose spatial distribution creates some visual patterns defined in terms of granularity, directionality, and repetitiveness. There exists different approaches to extract and represent textures. They can be classified into *space-based*, *frequency-based* models, and *texture signatures* [20]. Next, some of these approaches are described.

Co-occurrence matrix [28] is one the most traditional techniques for encoding texture information. It describes spatial relationships among grey-levels in a image. A cell defined by the position (i, j) in this matrix registers the probability at which two pixels of gray levels i and j occur in two relative positions. A set of co-occurrence probabilities (such as, energy, entropy, contrast) has been proposed to characterize textured regions. Other example of space-based method includes the use of **Auto-Regressive Models** [29].

Frequency-based texture descriptors include, for instance, the Garbor wavelet coefficients [30]. An example of texture signatures can be found in the proposal of Tamura *et al.* [31]. This descriptor aims to characterize texture information in terms of contrast, coarseness, and directionality. The MPEG-7 initiative proposed three texture descriptors: texture browsing descriptor, homogeneous texture descriptor, and local edge histogram descriptor [24].

3.1.3 Shape Descriptors: In pattern recognition and related areas, shape is an important characteristic to identify and distinguish objects [32, 33].

Shape descriptors are classified into *boundary-based (or contour-based)* and *region-based* methods [33]. This classification takes into account whether shape features are extracted from the contour only or from the whole shape region. These two classes, in turn, can be divided into *structural (local)* and *global* descriptors. This subdivision is based on whether the shape is represented as a whole or represented by segments/sections. Another possible classification categorizes shape description methods into *spatial* and *transform* domain techniques, depending on whether direct measurements of the shape are used or a transformation is applied [34][2].

Next, we present a brief overview of some shape descriptors. More details about existing shape representation techniques can be found in [32, 33, 35, 36].

Moment Invariants: For Moment Invariants, each object is represented by a 14-dimensional feature vector, including two sets of normalized Moment Invariants [37, 38], one from the object contour and another from its solid object silhouette. Again, the Euclidean distance is usually used to measure the similarity between different shapes as represented by their Moment Invariants.

Curvature Scale Space (CSS) [39, 40]: The CSS descriptor is used in the MPEG-7 standard [41] and represents a multiscale organization of the curvature zero-crossing points of a planar curve. In this sense, the dimension of its feature vectors varies for different contours, thus a special matching algorithm is necessary to compare two CSS descriptors (e.g., [15]).

Beam Angle Statistics (BAS) [42]: The BAS descriptor is based on the *beams* originated from a contour pixel. A beam is defined as the set of lines connecting a contour pixel to the rest of the pixels along the contour. At each contour pixel, the angle between a pair

[2]Taxonomies of shape description techniques can be found in [33–35].

of lines is calculated, and the shape descriptor is defined by using the third-order statistics of all the beam angles in a set of neighborhoods. The similarity between two BAS moment functions is measured by an optimal correspondent subsequence (OCS) algorithm, as shown in [42].

Tensor Scale Descriptor (TSD) [43]: TSD is a shape descriptor based on the tensor scale concept [44]— a morphometric parameter yielding a unified representation of local structure thickness, orientation, and anisotropy. That is, at any image point, its tensor scale is represented by the largest ellipse (2D) centered at that point and within the same homogeneous region. TSD is obtained by extracting the tensor scale parameters for the original image and then computing the ellipse orientation histogram. TSDs are compared by using a correlation-based distance function.

Contour Saliences (CS) [45]: The CS computation uses the Image Foresting Transform [46] to compute the salience values of contour pixels and to locate salience points along the contour by exploiting the relation between a contour and its internal and external skeletons [47]. The contour salience descriptor consists of the salience values of salient pixels and their location along the contour, and on a heuristic matching algorithm as distance function.

Segment Saliences (SS) [45]: The segment salience descriptor is a variation of the contour salience descriptor which incorporates two improvements: the *salience values* of contour segments, in the place of salience values of isolated points, and another matching algorithm that replaces the heuristic matching by an optimum approach. The salience values along the contour are computed and the contour is divided into a predefined number s of segments of the same size. The internal and external influence areas of each segment are computed by summing up the influence areas of their corresponding pixels. In [45], SS is showed to present a better effectiveness than several other shape descriptors.

3.2 CBIR Systems

Several CBIR systems have been proposed recently. Even though a few of them became commercial products [3], many CBIR systems were proposed as research prototype, being developed in universities and research laboratories.

QBIC (*Query by Image Content*) by IBM [3], Photobook [48], developed by the Massachusetts Institute of Technology (MIT), Chabot, Netra, and VisualSEEK [49] allow query by image content. Next, some of them are described.

Chabot [1] integrates image content retrieving based on color information with text-based queries. Its interface allows user to search and update the image database. This system does not include texture and shape descriptors.

The QBIC (*Query by Image Content*) system was developed by IBM [3]. QBIC uses color, shape, and texture to retrieve image databases. Query specification follows the *query-by-example* paradigm. A user can sketch a shape, select colors, indicate color distributions, or pre-defined textures.

Ma *et al.* [50] describe a toolbox for browsing large database collections called *Netra*. This prototype uses color, texture, shape, and spatial location of image segmented regions to retrieve similar images from a database.

More recently, Cox *et al.* [51] present the PicHunter system. In this system, a Bayesian framework is used to model user needs during query formulation. With a different approach, [52] describes an image retrieval system based on regions of interest, that is, regions that contain relevant objects of a given image. Another region-based image retrieval system is the Blobworld [53]. In this system, pixels are clustered according to their color and texture properties. These clusters are supposed to represent the image content.

A more complete description of existing CBIR systems can be found in [54].

3.3 Content-Based Image Search Component (CBISC)

Another recently adopted solution for constructing CBIR systems relies on the use of components. Content-Based Image Search Component (CBISC) [55], for example, provides an easy-to-install search engine to query images by content. It can be readily tailored for a particular collection by a domain expert, who carries out a clearly defined set of pilot experiments. It supports the use of different types of vector-based image descriptors (metric and non-metric; color, texture, and shape descriptors; with different data structures to represent feature vectors), which can be chosen based on the pilot experiment, and then easily combined to yield improved effectiveness. Besides, it encapsulates a metric index structure [11] to speed up the search process, that can be easily configured for different image collections.

The *CBISC Open Digital Library (ODL) component* is an Open Archives Initiative (OAI)-like search component which aims at supporting queries on image content. As in the OAI protocol [56], queries are submitted via HyperText Transfer Protocol (HTTP) requests. However, this was generalized to an extended OAI (XOAI) protocol for image search, that fits into the ODL framework [57]. As it is typical with XOAI protocols, each request specifies the Internet host of the HTTP server and gives a list of key-value pairs. Two special requests ("verbs") are supported by this image search component:

1. **ListDescriptors:** This verb is used to retrieve the list of image descriptors supported by the *CBISC*. No arguments are required for this verb.

2. **GetImages:** This verb is used to retrieve a set of images by taking into account their contents. Required arguments specify the query image, the descriptors to be used, and the kind of query. The *CBISC* supports two kinds of queries: *K-nearest neighbor query (KNNQ)*

and *range query (RQ)*.

3.4 Indexing Structures

Not only does the effectiveness but also the efficiency (measured in terms of retrieving time) needs to be taken into account during the design of CBIR systems. Usually, fast searching strategies rely on the use of effective indexing schemes. However, as pointed out earlier, images usually are represented as points in high dimensional spaces. In this scenario, traditional indexing schemes (such as the approaches based on the *R-trees* [58]), which perform reasonably well for a small number of dimensions, have a poor performance. This phenomena is called the "curse of dimensionality". One of the commonly approaches used for addressing this problem is applying dimension reduction techniques, such as Principal Component Analysis (PCA), and then using a traditional multidimensional indexing structure.

Another important research area includes the investigation of **Metric Access Methods** (MAMs). MAM is a class of access method (AM) that is used to manage large volumes of metric data allowing insertions, deletions and searches [12]. The definition of these indexing approaches relies on the use of a metric space. A metric space is a pair (O, d), where O denotes the domain of a set of objects $O = (O_1, O_2, \ldots, O_n)$, and d is a metric distance with the following properties: (i) symmetry ($d(O_1, O_2)$ = $d(O_2, O_1)$), (ii) positiveness ($0 < d(O_1, O_2) < \infty$, $O_1 \neq O_2$ and $d(O_1, O_2) = 0$), and (iii) triangle inequality ($d(O_1, O_3) \leq d(O_1, O_2) + d(O_2, O_3)$). Examples of MAMs include, among others, the M-tree [11] and the Slim-tree [12].

Further details about multidimensional indexing structures can be found in [8, 9].

3.5 Effectiveness Measures

Image descriptors vary with the application domain and expert requirements. Thus, in order to identify appropriate image descriptors (used in extraction and distance computation algorithms), experts must perform a set of experiments to evaluate them in terms of effectiveness for a given collection of images. Effectiveness evaluation is a very complex task, involving questions related to the definition of a collection of images, a set of query images, a set of relevant images for each query image, and adequate retrieval effectiveness measures.

The evaluation of image descriptors and CBIR systems usually adopts the *query-by-example (QBE)* [7] paradigm. This paradigm, in the image retrieval context, is based on providing an image as input, extracting its visual features (e.g., contour saliences), measuring the distance between the query image and the images stored in the image database and, finally, ranking the images in increasing order of their distance from the query image (similarity).

Since each descriptor represents an image as a "point" in the corresponding metric space, its effectiveness will be higher as more separate the clusters of relevant images are in the metric space; and as more compact the clusters are in the metric space, higher will be the robustness of the image descriptor with respect to an increase in the number of classes. Therefore, a "good" effectiveness measure should capture the concept of *separability*, and perhaps the concept of *compact-ability* for sake of robustness. More formally, the compact-ability of a descriptor indicates its invariance to the object characteristics that belong to a same class, while the separability indicates its discriminatory ability among objects that belong to distinct classes [15]. While these concepts are commonly used to define validity measures in cluster analysis [59, 60], they seem to not have caught much attention in the literature of CBIR systems, where one of the most used effectiveness measures is *Precision* $\times$ *Recall* [61].

Precision vs. Recall ($P \times R$) curve is the commonest evaluation measure used in CBIR domain. Precision is defined as the fraction of retrieved images which is relevant to a query. In contrast, recall measures the fraction of the relevant images which has been retrieved. A recall is a non-decreasing function of rank, while precision can be regarded as a function of recall rather than rank. In general, the curve closest to the top of the chart indicates the best performance.

The effectiveness in image retrieval was discussed with respect to the Precision$\times$Recall measure in [45], where the multiscale separability [15] was proposed as a more appropriate effectiveness measure.

Examples of other effectiveness measures include the $\theta \times$ recall curve [25], average precision [61], and average normalized modified retrieval rank (ANMRR) [24].

4 User Interaction in CBIR Systems

From the user's perspective, CBIR systems offer more flexibility in specifying queries than those based on metadata. On the other hand, they present new challenges. The first is how to help users in the *query specification* process. Another problem is *information overload* – how to present the result to the user in a meaningful way. A third issue is that of providing users with tools to *interact* with the system in order to refine their query. This section presents a brief overview of existing approaches that address these problems.

4.1 Query Specification

Several querying mechanisms have been created to help users define their information need. Asladogan *et al.* [7] presented a list of possible query strategies that can be employed in CBIR systems. This list includes, for example, *simple visual feature query*, *feature combination query*, *localized feature query*, *query by example*, *user-defined attribute query*, *object relationship query*, and *concept queries*. For instance, in the case of a feature combination query, a user could ask the system to *"Retrieve images with blue color and stripped texture, where both properties have the same weight"*.

Another distinction is made based on whether the user is looking for a class of similar items to a given query pattern (*"category search"*) or is looking for a particular target item (*"target search"*) [62].

4.2 Result Visualization

The most common result presentation technique is based on showing a 2D (two-dimensional) grid of thumbnail (miniature) image versions [1, 3]. The grid is organized according to the similarity of each returned image with the query pattern (e.g. from left to right, from top to bottom). It is a $n \times m$ matrix, where position $(1, 1)$ is occupied by a thumbnail of the query pattern, position $(1, 2)$ by the one most similar to it, and so on. This helps browsing, allowing users to simply scan the grid image set as if they were reading a text [63]. This approach, however, displays retrieved images of different similarity degrees at the same physical distance from the image query: e.g., images $(1, 2)$ and $(2, 1)$ are displayed at the same physical distance from the query pattern, but the former is more similar to it than the latter. [64] and [65] try to improve this visual structure by studying zoom properties to enhance image browsing. Rodden *et al.* [63], in turn, investigate whether it benefits users to have sets of thumbnails arranged according to their similarity, so images that are alike are placed together. They describe experiments to examine whether similarity-based arrangements of the candidate images help in picture selection.

Other display approaches try to consider relative similarity not only between the query pattern and each retrieved image, but also among all retrieved images themselves [66, 67]. These initiatives have the drawback that visually similar images which are placed next to each other can sometimes appear to merge or overlap, making them less eye-catching than if they were separated [63].

Stan *et al.* [67] describe an exploration system for an image database, which deals with a tool for visualization of the database at different levels of details based on a multi-dimensional scaling technique. This visualization technique groups together perceptual similar images in a hierarchy of image clusters. Retrieved images can overlap. The overlap problem is also found in El Niño image database [66]. In this context, Tian *et al.* [68] propose a PCA (Principal Component Analysis)-based image browser which looks into an optimization strategy to adjust the position and size of images in order to minimize overlap (maximize visibility) while maintaining fidelity to the original positions which are indicative of mutual similarities.

Torres et al. present in [19] two visualization techniques based on Spiral and Concentric Rings to explore query results (see Figure 4). These visual structures are centered on keeping user focus on the query image and on the most similar retrieved images. These strategies improve traditional 2D grid presentation and avoid image overlaps, commonly found in CBIR systems.

4.3 Relevance Feedback

Relevance feedback (RF) is a commonly accepted method to improve the effectiveness of retrieval systems interactively [69]. Basically, it is composed of three steps: (a) an initial search is made by the system for a user-supplied query pattern, returning a small number of images; (b) the user then indicates which of the retrieved images are useful (relevant); (c) finally, the system automatically reformulates the original query based upon user's relevance judgments. This process can continue to iterate until the user is satisfied. RF strategies help to alleviate the semantic gap problem, since it allows the CBIR system to learn user's image perceptions. RF strategies usually deal with small training samples (typically less than 20 per round of interaction), asymmetry in training sample (a few negative examples are usually fed back to the system), and real time requirement (RF algorithms should be fast enough to support real-time user interaction) [62]. Another important issue is concerned with the design and implementation of learning mechanisms. The commonest strategies use *weight-based learning approaches* [70], *genetic algorithms* [71], *Bayesian probabilistic methods* [51], and *Support Vector Machines* [72].

5 Applications

The CBIR technology has been used in several applications such as fingerprint identification, biodiversity information systems, digital libraries, crime prevention, medicine, historical research, among others. Some of these applications are presented in this section.

5.1 Medical Applications

The use of CBIR can result in powerful services that can benefit biomedical information systems. Three large domains can instantly take advantage of CBIR techniques: teaching, research, and diagnostics [73]. From the teaching perspective, searching tools can be used to find important cases to present to students. Research also can be enhanced by using services combining image content information with different kinds of data. For example, scientists can use mining tools to discover unusual patterns among textual (e.g., treatments reports, and patient records) and image content information. Similarity queries based on image content descriptors can also help the diagnostic process. Clinicians usually use similar cases for case-based reasoning in their clinical decision-making process. In this sense, while textual data can be used to find images of interest, visual features can be used to retrieve relevant information for a clinical case (e.g., comments, related literature, HTML pages, etc.).

5.2 Biodiversity Information Systems

Biologists gather many kinds of data for biodiversity studies, including spatial data, and images of living beings. Ideally, Biodiversity Information Systems (BIS) should help researchers to enhance or complete their knowledge and understanding about species and their habitats by combining textual, image content-based, and geographical queries. An example of such a query might start by providing an image as input (e.g., a photo of a fish) and then asking the system to *"Retrieve all database images containing fish whose fins are shaped like those of the fish in this photo"*. A combination of this query with textual and spatial predicates would consist of *"Show the drainages where the fish species with 'large eyes' coexists with fish whose fins are shaped like those of the fish in the photo"*. Examples of initiatives in this area include [55, 74].

5.3 Digital Libraries

There are several digital libraries that support services based on image content [74–79]. One example is the digital museum of butterflies [74], aimed at building a digital collection of Taiwanese butterflies. This digital library includes a module responsible for content-based image retrieval based on color, texture, and patterns. In a different image context, Zhu *et al.* [76] present a content-based image retrieval digital library that supports geographical image retrieval. The system manages air photos which can be retrieved through texture descriptors. Place names associated with retrieved images can be displayed by cross-referencing with a Geographical Name Information System (GNIS) gazetter. In this same domain, Bergman *et al.* describe an architecture for storage and retrieval of satellite images and video data from a collection of heterogeneous archives. Other initiatives cover different concepts of the CBIR area. For example, while research presented in [77, 78] concentrates on new searching strategies for improving the effectiveness of CBIR systems, another popular focus is on proposing image descriptors [79].

6 Research Challenges

The implementation of CBIR systems raises several research challenges, such as:

- Formalisms need to be created to describe image content descriptions and related services. This formalism can guide the design and implementation of new applications based on image content.
- Not many techniques are available to deal with the semantic gap presented in images

and their textual descriptions. New tools for marking/annotating images (and their regions) need to be developed. Better semantically enriched descriptions can be created by taking advantage of ontologies [80, 81]. Another possible investigation area would be to incoporate classification strategies into the image retrieval process. The idea is to apply image retrieval and then classify the resulting images to change their order. In this case, the classifier works as an automatic approach for relevance feedback.

- Need for tools that automatically extract semantic features from images: extract high-level concepts contained in multimedia data.
- Development of new data fusion algorithms to support text-based and content-based retrieval combining information of different heterogeneous formats [17].
- Finding new connections, and mining patterns. Text mining techniques might be combined with visual-based descriptions.
- New user interfaces for annotating, browsing, and searching based on image content need to be investigated. Research in this area will require usability studies with practitioners.

In May 2006, the Brazilian Computing Society organized a workshop in São Paulo, Brazil, to identify *grand research challenges in computing* for the period 2006-2016. One of the five challenges is *Managing Information in Large Collections of Distributed Multimedia Data* [82]. This challenge includes, among others, scientific problems related to the design and implementation of content descriptors as well as extraction and indexing algorithms, use of dynamic and distributed indexing structures in peer-to-peer networks, and the study of new information visualization techniques and interfaces.

7 Conclusions

This paper has presented a brief overview of content-based image retrieval area. Firstly, we have presented a set of constructs aiming to define precisely the main related concepts. Next, we have described the main issues that need to be taken into account when designing this kind of image retrieving system: definition of appropriate image descriptors, feature vector representation and indexing, interaction mechanisms, among others.

8 Acknowledgments

This work is funded in part by CNPq, CAPES, FAPESP, FAEPEX, and by a Microsoft eScience grant.

9 References

[1] V. E. Ogle and M. Stonebraker. Chabot: Retrieval from Relational Database of Images. *IEEE Computer*, 28(9):40–48, Sep 1995.

[2] H. Lieberman, E. Rosenzweig, and P. Singh. Aria: An Agent for Annotating and Retrieving Images. *IEEE Computer*, 34(7):57–62, 2001.

[3] M. Flickner, H. Sawhney, W. Niblack, Q. Huang J. Ashley, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by Image and Video Content: the QBIC System. *IEEE Computer*, 28(9):23–32, Sep 1995.

[4] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the End of the Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, December 2000.

[5] Y. Rui, T. S. Huang, and S. F. Chang. Image Retrieval: Current Techniques, Promising Directions, and Open Issues. *Journal of Communications and Image Representation*, 10(1):39–62, March 1999.

[6] M. Das, R. Manmatha, and E. M. Riseman. Indexing Flower Patent Images Using Domain Knowledge. *IEEE Intelligent Systems*, pages 24–33, September/October 1999.

[7] Y. A. Aslandogan and C. T. Yu. Techniques and Systems for Image and Video Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):56–63, January/February 1999.

[8] V. Gaede and O. Gunther. Multidimensional Access Methods. *ACM Computing Surveys*, 30(2):170–231, 1998.

[9] C. Bohm, S. Berchtold, and D. A. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys (CSUR)*, 33(3):322–373, 2001.

[10] R. Weber, H. Schek, and S. Blott. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In *VLDB'98, Proceedings of 24rd International Conference on Very Large Data Bases*, pages 194–205, 1998.

[11] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *Proceedings of 23rd International Conference on Very Large Data Bases*, pages 426–435, Athens, Greece, 1997.

[12] C. Traina, B. Seeger, C. Faloutsos, and A. Traina. Fast Indexing and Visualization of Metric Datasets Using Slim-Trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):244–260, March/April 2002.

[13] M. A. Goncalves, E. A. Fox, and N. A. Kipp. Streams, Structures, Spaces, Scenarios, Societies (5S): A Formal Model for Digital Libraries. *ACM Transactions on Information Systems (TOIS)*, 22(2):1–43, 2004.

[14] M. Swain and D. Ballard. Color Indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[15] R. da S. Torres, A. X. Falcão, and L. da F. Costa. A Graph-based Approach for Multiscale Shape Analysis. *Pattern Recognition*, 37(6):1163–1174, June 2004.

[16] E. Persoon and K. Fu. Shape Discrimination Using Fourier Descriptors. *IEEE Transanctions on Systems, Man, and Cybernetics*, 7(3):170– 178, 1977.

[17] R. da S. Torres, A. X. Falcão, M. A. Goncalves, B. Zhang, W. Fan, E. A. Fox, and P. P. Calado. A New Framework to Combine Descriptors for Content-based Image Retrieval. In *Proc. of the 14th ACM Conference on Information and Knowledge Management*, pages 335–336, 2005.

[18] R. Shen, N. S. Vemuri, W. Fan, R. da S. Torres, and Edward A. Fox. Exploring Digital Libraries: Integrating Browsing, Searching, and Visualization. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital libraries*, pages 1–10, Chapel Hill, NC, USA, June 2006.

[19] R. da S. Torres, C. G. da Silva, C. B. Medeiros, and H. V. da Rocha. Visual Structures for Image Browsing. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pages 167–174, New Orleans, LA, USA, November 2003.

[20] A. del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1999.

[21] M. A. Stricker and M. Orengo. Similarity of Color Images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 381–392, 1995.

[22] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image Indexing Using Color Correlograms. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 762–768, Puerto Rico, June 1997.

[23] G. Pass, R. Zabih, and J. Miller. Comparing Images Using Color Coherence Vectors. In *Proceedings of the fourth ACM international conference on Multimedia*, pages 65–73, 1996.

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

[25] R.O. Stehling, M.A. Nascimento, and A.X. Falcão. A Compact and Efficient Image Retrieval Approach Based on Border/Interior Pixel Classification. In *Proceedings of the 11th ACM International. Conference on Information and Knowledge Management*, pages 102–109, McLean, Virginia, USA, November 2002.

[26] MPEG-7. http://ipsi.fhg.de/delite/projects/mpeg7/ (as of march 2003).

[27] S. F. Chang, T. Sikora, and A. Puri. Overview of the MPEG-7 Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):688–695, June 2001.

[28] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernatics*, 3(6):610–621, 1973.

[29] A. Khotanzad and R. L. Kashyap. Feature Selection for Texture Recognition Based on Image Synthesis. *IEEE Transactions on Systems, Man and Cybernatics*, 17(6):1087–1095, 1987.

[30] B. S. Manjunath and W. Y. Ma. Texture Features for Browsing and Retrieval of Image Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, August 1996.

[31] H. Tamura, S. Mori, and T. Yamawaki. Textural Features Corresponding to Visual Perceptron. *IEEE Transactions on Systems, Man and Cybernatics*, 8(6):460–473, 1978.

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

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

[34] M. Safar, C. Shahabi, and X. Sun. Image Retrieval by Shape: A Comparative Study. In *IEEE International Conference on Multimedia and Expo (I)*, pages 141–144, 2000.

[35] L.F. Costa and R.M. Cesar Jr. *Shape Analysis and Classification: Theory and Practice*. CRC Press, Boca Raton, FL, USA, 2001.

[36] B. M. Mehtre, M. S. Kankanhalli, and W. F. Lee. Shape Measures for Content Based Image Retrieval: A Comparison. *Information Processing and Management*, 33(3):319–337, 1997.

[37] M. K. Hu. Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.

[38] S. A. Dudani, K. J. Breeding, and R. B. McGhee. Aircraft Identification by Moment Invariants. *IEEE Transactions on Computers*, 26(1):39–45, January 1977.

[39] S. Abbasi, F. Mokhtarian, and J. Kittler. Enhancing CSS-based Shape Retrieval for Objects with Shallow Concavities. *Image and Vision Computing*, 18(3):199–211, February 2000.

[40] F. Mokhtarian and S. Abbasi. Shape Similarity Retrieval Under Affine Transforms. *Pattern Recognition*, 35(1):31–41, January 2002.

[41] M. Bober. MPEG-7 Visual Shape Descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):716–719, June 2001.

[42] N. Arica and F. T. Y. Vural. BAS: A Perceptual Shape Descriptor Based on the Beam Angle Statistics. *Pattern Recognition Letters*, 24(9-10):1627–1639, June 2003.

[43] P. Miranda, R. da S. Torres, and A. X. Falcão. TSD: A Shape Descriptor Based on a Distribution of Tensor Scale Local Orientation. In *XVIII Brazilian Symposium on Computer Graphics and Image Processing*, pages 139–146, Natal, RN, Brazil, October 2005.

[44] P.K. Saha. Tensor Scale: A Local Morphometric Parameter With Applications to Computer Vision and Image Processing. Technical Report 306, Medical Image Processing Group, Department of Radiology, University of Pennsylvania, September 2003.

[45] R. da S. Torres and Alexandre X. Falcão. Contour Salience Descriptors for Effective Image Retrieval and Analysis. *Image and Vision Computing*, 2006. To appear.

[46] A. X. Falcão, J. Stolfi, and R. A. Lotufo. The Image Foresting Transform: Theory, Algorithms, and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29, Jan 2004.

[47] M. Leyton. Symmetry-Curvature Duality. *Computer Vision, Graphics, and Image Processing*, 38(3):327–341, 1987.

[48] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based Manipulation of Image Databases. In *SPIE Storage and Retrieval for Image and Video Databases II*, pages 34–47, San Jose, CA, 1994.

[49] J. R. Smith and S. F. Chang. VisualSEEk: A fully automated content-based image query system. In *Proceedings of the ACM Multimedia*, pages 87–98, Boston, MA, November 1996.

[50] W. Y. Ma and B. S. Manjunath. Netra: A Toolbox for Navigating Large Image Databases. In *IEEE International Conference on Image Processing*, pages 256–268, 1997.

[51] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, January 2000.

[52] K. Vu, K. A. Hua, and W. Tavanapong. Image Retrieval Based on Regions of Interest. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):1045–1049, July/August 2003.

[53] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image Segmentation Using Expectation-Maximization and its Application to Image Querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, August 2002.

[54] R. C. Veltkamp and M. Tanase. Content-Based Image Retrieval Systems: A Survey. Technical Report UU-CS-2000-34, Institute of Information and Computing Sciences, Utrecht University, 2000.

[55] R. S. Torres, C. B. Medeiros, M. A. Goncalves, and E. A. Fox. A Digital Library Framework for Biodiversity Information Systems. *International Journal on Digital Libraries*, 6(1):3–17, February 2006.

[56] C. Lagoze and H. V. de Sompel. The Open Archives Initiative: Building a Low-Barrier Interoperability Framework. In *Proceedings of the Joint Conference on Digital Libraries*, pages 54–62, Roanoke, VA, USA, June 2001.

[57] H. Suleman, E. A. Fox, R. Kelapure, A. Krowne, and M. Luo. Building Digital Libraries from Simple Building Blocks. *Online Information Review*, 27(5):301–310, 2003.

[58] R. Guttman. R-Tree: A Dynamic Index to Structure for Spatial Searching. In *SIGMOD Conf. Ann. Meeting*, pages 47–57, Boston, 1984.

[59] J. Dunn. Well Separated Clusters and Optimal Fuzzy Partitions. *Cybernetics*, 4:95–104, 1974.

[60] D. Davies and D. Bouldin. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

[61] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc, Boston, MA, USA, 1999.

[62] X. S. Zhou and T. S. Huang. Relevance Feedback in Image Retrieval: A Comprehensive Review. *Multimedia Systems*, 8:536–544, 2003.

[63] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Does Organization by Similarity Assist Image Browsing? In *ACM Conference on Human Factors in Computing Systems*, volume 3, pages 190–197, 2001.

[64] B. B. Bederson. Photomesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 71–80, Orlando, FL, USA, 2001.

[65] T. T. A. Combs and B. B. Bederson. Does Zooming Improve Image Browsing? In *Proceedings of the Fourth ACM International Conference on Digital Libraries*, pages 130–137, Berkeley, California, USA, 1999.

[66] S. Santini, A. Gupta, and R. Jain. Emergent Semantics through Interaction in Image Databases. *IEEE Transactions on Knowledge and Data Engineering*, 13(3):337–351, May/June 2001.

[67] D. Stan and I. K. Sethi. eID: a System for Exploration of Image Databases. *Information Processing and Management*, 39(3):335–365, 2003.

[68] Q. Tian, B. Moghaddam, and T. S. Huang. Display Optimization for Image Browsing. In *Proceedings of the 2nd International Workshop on Multimedia Databases and Image Communications*, Amalfi, Italy, Sep 2001.

[69] S. D. MacArthur, C. E. Brodley, A. C. Kak, and L. S. Broderick. Interactive Content-Based Image Retrieval Using Relevance Feedback. *Computer Vision and Image Understanding*, 88(2):55–75, 2002.

[70] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. A Power Tool in Interactive Content-Based Image Retrieval. *IEEE Tran on Circuits and Systems for Video Technology*, 8(5):644–655, 1998.

[71] Cristina Lopez-Pujalte, Vicente P. Guerrero Bote, and Felix de Moya Anegon. Order-based fitness functions for genetic algorithms applied to relevance feedback. *Journal of the American Society for Information Science and Technology*, 54(2):152–160, 2003.

[72] S. Tong and E. Y. Chang. Support vector machine active learning for image retrieval. In *Proceedings of 9th ACM international conference on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM Press.

[73] H. Muller, N. Michoux, D. Bandon, and A. Geissbuhler. A Review of Content-Based Image Retrieval Systems in Medical Applications – Clinical Benefits and Future Directions. *International Journal of Medical Informatics*, 73(1):1–23, Feb 2004.

[74] J. S. Hong, H.Y. Chen, and J. Hsiang. A Digital Museum of Taiwanese Butterflies. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pages 260–261, San Antonio, Texas, United States, 2000. ACM Press.

[75] L. D. Bergman, V. Castelli, and C. Li. Progressive Content-Based Retrieval from Satellite Image Archives. *D-Lib Magazine*, 3(10), October 1997.

[76] B. Zhu, M. Ramsey, and H. Chen. Creating a Large-Scale Content-Based Airphoto Image Digital Library. *IEEE Transactions on Image Processing*, 9(1):163–167, January 2000.

[77] J. C. French, A. C. Chapin, and W. N. Martin. An Application of Multiple Viewpoints to Content-based Image Retrieval. In *Proceedings of the 3rd ACM/IEEE-CS joint conference on digital libraries*, pages 128–130, Washington, DC, USA, 2003.

[78] J. Z. Wang and Y. Du. Scalable integrated region-based image retrieval using irm and statistical clustering. In *Proceedings of the 1st ACM/IEEE-CS joint conference on digital libraries*, pages 268–277, 2001.

[79] Y. Wang, F. Makedon, J. Ford, L. Shen, and D. Goldin. Generating fuzzy semantic metadata describing spatial relations from images using the r-histogram. In *Proceedings of the 4th ACM/IEEE-CS joint conference on digital libraries*, pages 202–211, 2004.

[80] M. Addis, M. Boniface, S. Goodall, P. Grimwood, S. Kim, P. Lewis, K. Martinez, and A. Stevenson. SCULPTEUR: Towards a New Paradigm for Multimedia Museum Information Handling. In *Proc. of Semantic Web ISWC 2870*, pages 582–596, 2003.

[81] R. B. Freitas and R. da S. Torres. Ontosaia: Um ambiente baseado em ontologias para recuperacao e anotacao semi-automatica de imagens. In *Primeiro Workshop de Bibliotecas Digitais*, pages 60–69, Uberlandia. MG, Brasil, 2005.

[82] SBC. Grandes Desafios da Pesquisa em Computação no Brasil – 2006–2016, http://www.sbc.org.br, May 2006.