UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Universitat Politècnica de Catalunya
Enginyeria Tècnica Superior de Telecomunicacions
Image and Video Processing Group

Image-Based Query by Example Using MPEG-7 Visual Descriptors

Degree's Final Project Dissertation

by Carles Ventura Royo

Advisors:
Ferran Marquès Acosta
Jordi Pont Tuset

Barcelona, March 2010

Abstract

This project presents the design and implementation of a Content-Based Image Retrieval (CBIR) system where queries are formulated by visual examples through a graphical interface. Visual descriptors and similarity measures implemented in this work followed mainly those defined in the MPEG-7 standard although, when necessary, extensions are proposed. Despite the fact that this is an image-based system, all the proposed descriptors have been implemented for both image and region queries, allowing the future system upgrade to support region-based queries. This way, even a contour shape descriptor has been developed, which has no sense for the whole image. The system has been assessed on different benchmark databases; namely, MPEG-7 Common Color Dataset, and Corel Dataset. The evaluation has been performed for isolated descriptors as well as for combinations of them. The strategy studied in this work to gather the information obtained from the whole set of computed descriptors is weighting the rank list for each isolated descriptor.

Acknowledgments

I would like to express my greatest appreciation to my advisors, Ferran Marquès and Jordi Pont, for their support, stimulating suggestions and encouragement throughout the course of this research work.

I would also like to thank all my colleagues and friends for their help and useful suggestions, in particular Xavier Giró, and Albert Gil. Moreover, I am very grateful to my parents and my girlfriend for cheering me up constantly.

Contents

List of Figures

Chapter 1

Introduction

As a consequence of recent technology development in several fields, large image databases have been created. In this context, well organized databases and efficient storing and retrieval algorithms are absolutely necessary.

Most traditional methods of image retrieval consists in adding metadata, such as keywords or textual descriptions, to the images so that retrieval can be performed over the annotations. It is rather an efficient method for simple and small image databases, since only a few keywords are necessary to describe them. When the amount of images in the database grew and became more complex, researchers realized that the images having rich contents could not be described by only several semantic words. Furthermore, the keywords were very dependent on the observer.

These problems with traditional methods of image indexing have led to the rise of interest in techniques for retrieving images on the basis of automatically-derived features such as colour, texture, and shape, a technology now generally known as Content-Based Image Retrieval (CBIR). The earliest use of the term CBIR in the literature seems to have been by Kato in 1992 in order to describe his experiments [HK92]. Query by Example (QbE) is the most popular query technique, which involves providing the CBIR system with an example image. From this image, some visual features are extracted and compared to the features of each image of the database in order to retrieve the most similar ones.

The basic problem in CBIR is the gap between the high-level semantic concepts used by humans and the low-level visual features extracted from images by a computer to describe the images in a database. Two most important research topics in CBIR are (*i*) feature selection and measures of similarity, (*ii*) the techniques for ranking the images, i.e., how to choose the images that have to be displayed to the user. These two subjects have been studied in the current research work.

First, in Chapter 2, the state of the art is analyzed, describing some CBIR systems available in the literature and introducing a graphical user interface. Then, Chapter 3 presents the visual features proposed by the MPEG-7 standard. The similarity measures for each of these features are given in Chapter 4. Then, Chapter 5 presents a technique for selecting in which order the images have to be displayed to the user. In Chapter 6, the experimental results obtained are given and discussed. Conclusions and future lines of work are detailed in Chapter 7.

Chapter 2

State of the Art

The digitization of the media along with the reduction in the price of data storage has lead to the generalization of huge multimedia databases. The necessity of dedicated methods for indexing and retrieval the contents came up in order to access or make use of such databases in an efficient way. Content-Based Image Retrieval (CBIR) is a crucial factor for digital image storage, but it is still in a preliminary stage of development. Features such as color, texture, and shape are commonly used for retrieval. Section 2.1 provides an overview of some famous CBIR systems. Section 2.2 presents MIRROR, a CBIR system based on MPEG-7 visual descriptors which will be our reference. Finally, Section 2.3 presents GOS, a graphical user interface which will be used for our implemented search engine.

2.1 An overview of some CBIR systems

Although MIRROR is the most similar CBIR system to the one implemented by us, following we provide an overview of other representative systems [RHC99] in image retrieval and expose their characteristics briefly.

- QBIC (Query By Image Content) [FSN^{+}97, NBE^{+}93] was the first commercial CBIR system. It was developed by IBM in the first half of the 90s. It consists in a search engine that sorts through database images according to colors, textures, shapes, sizes, and their positions. Its structure and techniques used have made a great effect on most of the later image retrieval systems. QBIC supports queries based on example images, user-constructed sketches and drawings, selected color and texture patterns, camera and object motion, and other graphical information. Any resulting image can be used as a new query in order to improve the search. Moreover, text-based keyword search can be combined with content-based similarity search. QBIC also takes into account the high-dimensional feature indexing. The main aim of IBM was to design a good search engine to the detriment of designing a more attractive web-like interface.

- Virage [BFG96] is a CBIR system which was developed at Virage Inc. This system, like QBIC, supports visual queries based on color, composition (color layout), texture, and shapes. The main difference between these two search engines is that Virage also supports arbitrary combinations of the above four atomic queries. Thus, this system allows the user to adjust the weights associated with these atomic features.

- RetrievalWare [EXC98] is a content-based image retrieval engine developed by Excalibur Technologies Corporation. It was commercially launched in 1992. Its emphasis was in neural nets to image retrieval. Its more recent search engines uses features such as color, shape, texture, brightness, color layout, and aspect ratio of the image. Similar to Virage, it also supports the combination of these features and their associated weights can be adjusted by the user. Like QBIC, Excalibur Technologies Corporation focused on the search engine to the detriment of designing a good graphical interface.

- Photobook [PPS95] is a computer system that allows the user to browse image databases, using both text annotation information and by having the computer searching the images directly based on their content. This search engine, which was developed at the MIT Media Lab, consists of three subbooks from which shape, texture, and face features are extracted, respectively. It can also handle combinations of these descriptors. In its more recent version, humans are involved in the image annotation and retrieval loop, like in the MIRROR system. The human factor was incorporated because there was no single feature which can best model image from each and every domain.

- VisualSEEk [SC96] is a content-based search engine developed at Columbia University. The visual features used in this system are color set and wavelet transform-based texture feature. It also extended local queries at the region level. Therefore, VisualSEEk supports queries based on spatial properties, such as size, location and relationships to other regions. It was the first search engine which allowed users to flexibly query for images by specifying both visual features and spatial properties of the desired images. The user interface let the formulation of queries under the form of sketches by using a graphical editor. This enables the user to submit a sunset query as red-orange color region on top and blue or green region at the bottom as its sketch. VisualSEEk was developed following a server-client architecture on the World Wide Web (WWW), where the client was a Java application and the server contained a test-bed of 12,000 images. The server side of the system was extended to the WebSEEk [SC97], which is a web-oriented search engine.

- NeTra [MM97] is a prototype image retrieval system that uses color, texture, shape, and spatial location information in the segmented image regions. It was developed in the UCSB Alexandria Digital Library (ADL) project. A distinguishing aspect of this system is its incorporation of a robust automated image segmentation algorithm that allows object or region-based search. Another important component of the system includes an efficient color representation, and indexing of color, texture, and shapes features for fast search and retrieval.

- MARS (Multimedia Analysis and Retrieval System) [HMR96] was developed at University of Illinois at Urbana-Champaign. It was the first system which incorporated feedback relevance techniques in order to emulate the users' needs. It was an interdisciplinary research effort involving multiple research communities such as computer vision, database management system, and image retrieval. The searching criteria can be specified by the user using either an image example or selected colors and texture patterns. Regarding the feedback techniques, next to each resulting image a slider allows the user to specify their relevance. Moreover, each image has a checkbox so that the user can mark which images are the most similar to the expected results. The system takes into account all this information intoduced by the user in order to improve the image retrieval results. The graphical interface offered by MARS supposed an evolution of the information visual treatment in the screen using a window-like design.

- ImageGrouper [NMH02] is a CBIR system based in querying groups of images which substituted MARS. Both search engines were developed by the same responsibles. Its main innovation is the way in which the retrievals are done with a technique called Query-by-Groups. It allows the user to classify all the resulting images in three groups: (*i*) positive (images that fulfill the user's expectations), (*ii*) negative (images that do not fulfill the user's expectations), and (*iii*) neutral.

- SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries) [WLW00] was develop at Stanford University between 1999 and 2000. It is an image retrieval system that uses semantics classification methods, a wavelet-based approach for feature extraction, and integrated region matching based upon image segmentation. The images are represented by a set of regions which are characterized by color, texture, shape, and location. The system classifies images into semantic categories, such as texture or non-textured, graph or photograph. A measure for the overall similarity between images, called Integrated Region Matching, is developed using a region-matching scheme that integrates properties of all the regions in the images, resulting in a simple querying interface.

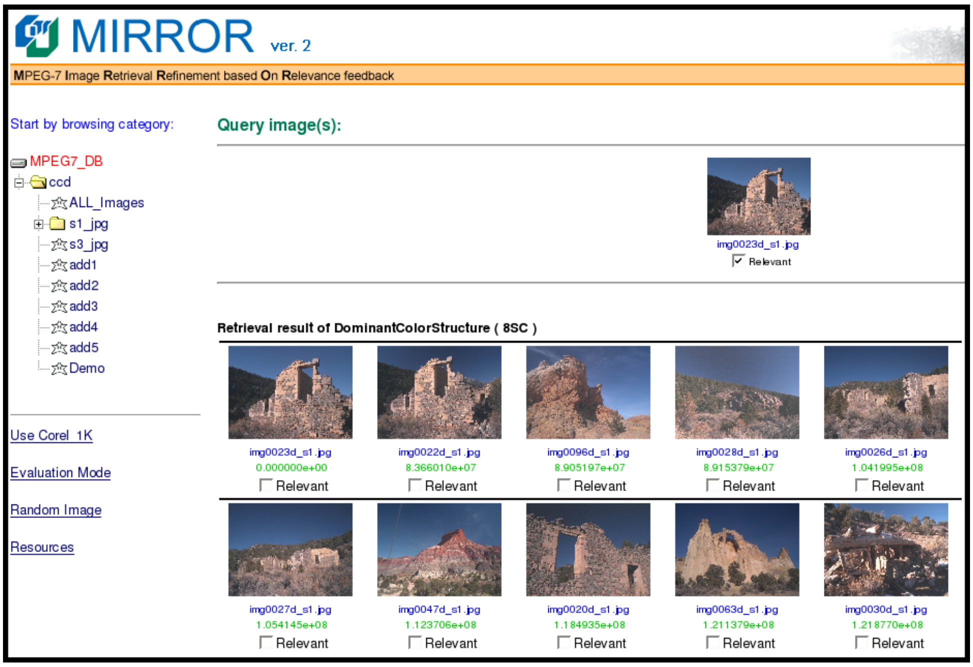
- CuZero [ZC08] is a search engine which employs a unique query that allows zero-latency query formulation. The response time was the key aspect aimed by this system, which was developed at Columbia University. It uses both text-based and content-based querying techniques. After users enter each single word, relevant visual concepts are automatically recommended in real time. CuZero also introduces a new intuitive visualization system. The use of two grids is proposed. In one of them, the recommended images are presented whereas in the other one a map with the used keywords and query images are showed. The recommended images change depending on the cursor's position in the map of the second grid. Therefore, this system allows the user to give more importance to some words or images in a more intuitive way.

- Multicolr [MUL] is a public site from Ide, a company that develops image identification and visual search software for business and industry. This site lets the user browse by color through a huge collection of over ten million licensed photographs. The interface is very simple. The user only has to click on up to ten colors as a selection and the fifty pictures that feature those colors are showed. These colors can be added or removed in an easy way in order to change the color selection.

- ImgSeek [IMG] is a free open source photo collection manager and viewer with content-based search. The query is given by either a sketch painted by the user or another supplied image. The searching algorithm makes use of multiresolution wavelet decomposition of the query and database images. It is also provided with an advanced keyword searching for metadata. Furthermore, this system allows the user to cluster the images automatically by similarity, such as color, date, filename, and image features. Wavelet algorithms, metric and query ideas are based on [JFS95].

2.2 MIRROR: the reference CBIR system

A content based image retrieval system, called MPEG-7 Image Retrieval Refinement based On Relevance feedback (MIRROR) [WCP05], is developed for evaluating MPEG-7 visual descriptors and developing new retrieval algorithms. It includes a web-based user interface for query by image example retrieval. Figure 2.1 shows a capture of MIRROR.



Query image(s):

img0023d_s1.jpg
☑ Relevant

Retrieval result of DominantColorStructure (8SC)

img0023d_s1.jpg 0.000000e+00 □ Relevant
img0022d_s1.jpg 8.366010e+07 □ Relevant
img0096d_s1.jpg 8.905197e+07 □ Relevant
img0028d_s1.jpg 8.915379e+07 □ Relevant
img0026d_s1.jpg 1.041995e+08 □ Relevant
img0027d_s1.jpg 1.054145e+08 □ Relevant
img0047d_s1.jpg 1.123706e+08 □ Relevant
img0020d_s1.jpg 1.184935e+08 □ Relevant
img0063d_s1.jpg 1.211379e+08 □ Relevant
img0030d_s1.jpg 1.218770e+08 □ Relevant

Figure 2.1: A capture of MIRROR

The application of current CBIR systems is limited and not satisfactory as the intrinsic gap between high-level concepts and low-level features is not considered in many CBIR systems. Users have to express their requirements in terms of low level image features. MIRROR is implemented to develop techniques to address this problem and supports various MPEG-7 visual descriptors (see Chapter 3) for representation and extraction of image features.

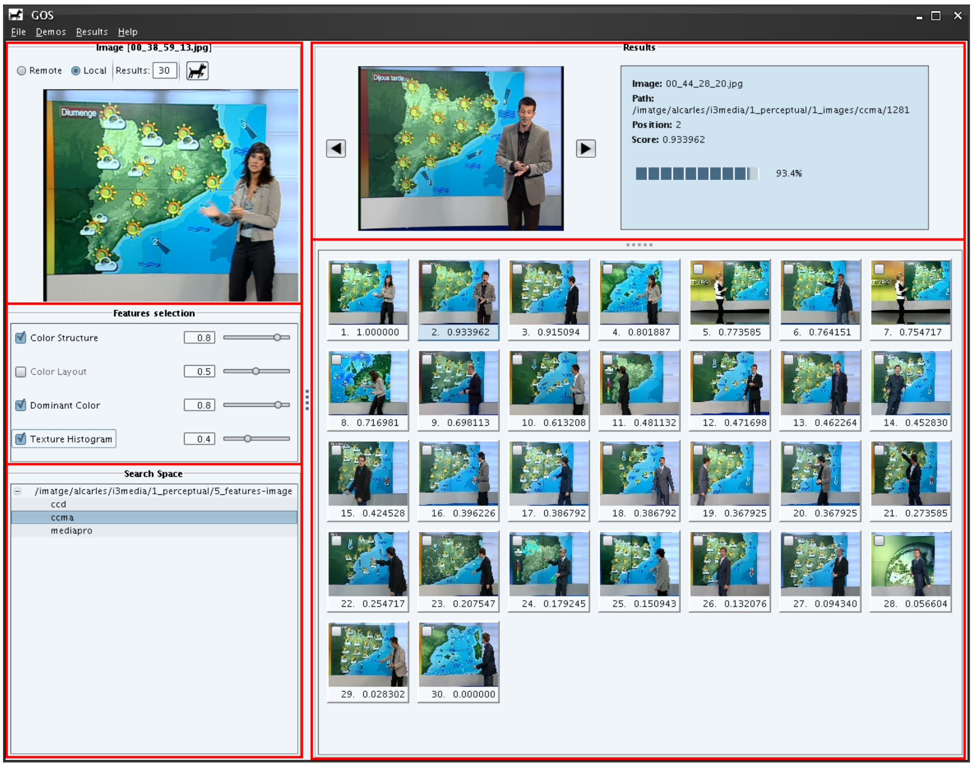
In addition, it uses Relevance Feedback (RF) techniques, which take into account the users' feedback during the retrieval process to effectively capture the high-level query and concepts. The users' information is used to dynamically update the weights given to low-level features. Successful application of relevance feedback techniques in CBIR system highly depends on the use of representative image features in the feedback process.

MIRROR supports MPEG-7 color descriptors (Dominant Color, Color Layout, Scalable Color and Color Structure) and one texture descriptor (Texture Edge Histogram) for CBIR similarity measures. Various matching tools are defined for different descriptors and description schemes.

2.3 GOS: The graphical user interface for our search engine

Although this project focuses on the search engine, we will take advantage of a graphical user interface, called GOS (Graphical Object Searcher), which is being developed by the Image and Video Processing Group of the Signal Theory and Communications Department of the Technical University of Catalonia (UPC). This interface manages the user interaction with the query by example system, allowing the user to retrieve some images similar to the query in an attractive and intuitive way. The first task of this tool is that users can introduce the set of configuration parameters required by the search engine. Afterwards, the GOS interface has to wait for the results returned by the retrieval system and then shows this retrieved content in a framework for navigation and browsing.

According to the Human-Computer Interaction principles, the way of showing information to users affects their behaviour. That is the reason why the interface has been designed following User-Centered Design in order to have an impact on the usability of the system and the final satisfaction degree. The window application is divided in two main areas: (*i*) the query panel on the left, which groups the query parameters, and (*ii*) the results panel on the right, which shows and manages the results. The upper part of the query panel always contains the query image. This part also includes a cell where the desired amount of results can be introduced, two radio buttons in order to choose between a local and a remot execution, and a pointing-dog icon which launches the search engine. The middle part of this panel allows the user to choose which visual descriptors have to be used and their weights. The lower part shows a tree whose nodes represent image collections and are selected by the user in order to specify the search space. Once a rank list of the top images in the search image and their associated distances is returned by the search engine, these images are shown in the results panel. Its lower part includes the thumbnails and the scores of the retrieved images. The upper part of the results panel shows a higher definition version of the selected result and detailed information about it, such as the image filename, the path, the position in the rank list, and the score obtained. All these parts explained have been marked in a capture of GOS which is shown in Figure 2.2.



Results: 30

Image: 00_44_28_20.jpg
Path: /imatge/alcarles/i3media/1_perceptual/1_images/ccma/1281
Position: 2
Score: 0.933962

☑ Color Structure 0.8
□ Color Layout 0.5
☑ Dominant Color 0.8
☑ Texture Histogram 0.4

/imatge/alcarles/i3media/1_perceptual/5_features-image
- ccd
- ccma
- mediapro

Figure 2.2: A capture of GOS program divided into its main parts

Chapter 3

Visual Descriptors

3.1 Introduction

As defined in [MP3] MPEG-7, also known as “Multimedia Content Description Interface,” provides a standardized set of technologies for describing multimedia content. The visual descriptors, which are specified in this standard, define the syntax and the semantic of each feature (meta-data element). These descriptors are classified according to the feature which is described, such as color, shape, texture, etc. Those related to color, such as *Dominant Color*, *Color Structure*, and *Color Layout*, all of which specified in the MPEG-7 standard, are presented in Section 3.2. Next, two descriptors related to texture are presented in Section 3.3. One of them, known as *Texture Edge Histogram*, is specified in the standard. The other one, which is based on the power in the Haar-Transform bands, does not belong to MPEG-7. Finally, the MPEG-7 *Contour-Shape Descriptor* is presented in Section 3.4 as an example of shape descriptors. Furthermore, some auxiliary implemented descriptors are introduced in each section. Some visual descriptors proposed by MPEG-7 has not been chosen for the following reasons. The *Scalable Color Descriptor* has as a main approach the storage efficiency, which is not our focus at the moment. The *Homogeneous Texture Descriptor* is quite effective in characterizing homogeneous texture regions. Therefore, we will implement it when our approach is to design a region-based search engine. The last one not considered is the *Region-Based Shape Descriptor* because it is based on both boundary and internal pixels. This descriptor is usually used in order to retrieve trademarks, which is not our goal.

In this work, the various descriptors will be analyzed from the viewpoint of the area of support and their implementation. With regard to the area of support, each descriptor can be computed both locally (on an image region) and globally (on the whole image). However, there are some descriptors which are more common to be computed only on the entire image.

In addition to that, given that we are working with a structure called *Binary Partition Tree* (BPT) (see Appendix A), a descriptor can be implemented in two different ways. On the one hand, there are some descriptors that are computed recursively. In other words, given the descriptors of two image regions, the descriptor of the region resulting of the fusion of these regions can be computed. In this case, the descriptor is said to be *bottom-up expandable*.

On the other hand, some descriptors cannot be computed recursively, so these are processed without taking into account their position in the BPT structure. In this case, no region is viewed as the result of merging its children's region, so all the region descriptors are computed the same way, like the BPT's leaves.

Finally, it has to be said that a new collaborative structure has been created. This structure takes advantage of the descriptors computed previously. For example, if one descriptor needs another one which has been already calculated, it is not necessary to recompute it.

3.2 Color Descriptors

Color is an important visual attribute for both human vision and computer processing [MPS02]. This section provides the reader with an overview of a selection of descriptors that are considered by the MPEG-7 group for describing visual content based on color. These can be computed in different color spaces, which are presented in Appendix B.

3.2.1 Dominant Color Descriptor

The *Dominant Color Descriptor* (DCD) allows a specification of a small number of representative color values in an image or image region, as well as their statistical properties, such as distribution and variance. Specifically, the DCD is defined in [MPS02] to be:

$$F = \{(\mathbf{c}_i, p_i, v_i), s\}, \qquad (i = 1, 2, ..., N) \tag{3.1}$$

where N is the number of dominant colors. Each dominant color value $\mathbf{c}_i$ is a vector of corresponding color space component values (e.g. a 3-D vector in the RGB color space). A maximum of eight dominant colors was found to be sufficient to represent an image or image region [MPS02]. The percentage p_i, which is normalized to a value between 0 and 1, is the fraction of pixels in the image or image region corresponding to color $\mathbf{c}_i$, and $\sum_i p_i = 1$. The optional color variance parameter v_i describes the variation of the color values of the pixels which have been assigned to the cluster C_i represented by the dominant color value $\mathbf{c}_i$. The spatial coherency s is a single number that represents the overall spatial homogeneity of the dominant colors in the image. This parameter is calculated in order to identify groups of pixels of the same dominant color that are spatially connected. The spatial coherency is a linear combination of the individual spatial coherence values with the corresponding percentages p_i being the weights. Each individual spatial coherency is computed using a four connectivity mask and obtaining which percentage of neighbors are assigned the same dominant color value as the pixel visited for the whole image or image region.

In order to extract the values of the dominant colors, two techniques have been implemented. The first one is specified in the MPEG-7 standard and is initialized with one cluster in the color space represented by the centroid (center of mass) of all pixels belonging to the whole image or image region. Then, the algorithm follows a sequence of splitting clusters and updating centroids until a stopping criterion (minimum distortion or maximum number of clusters) is reached.

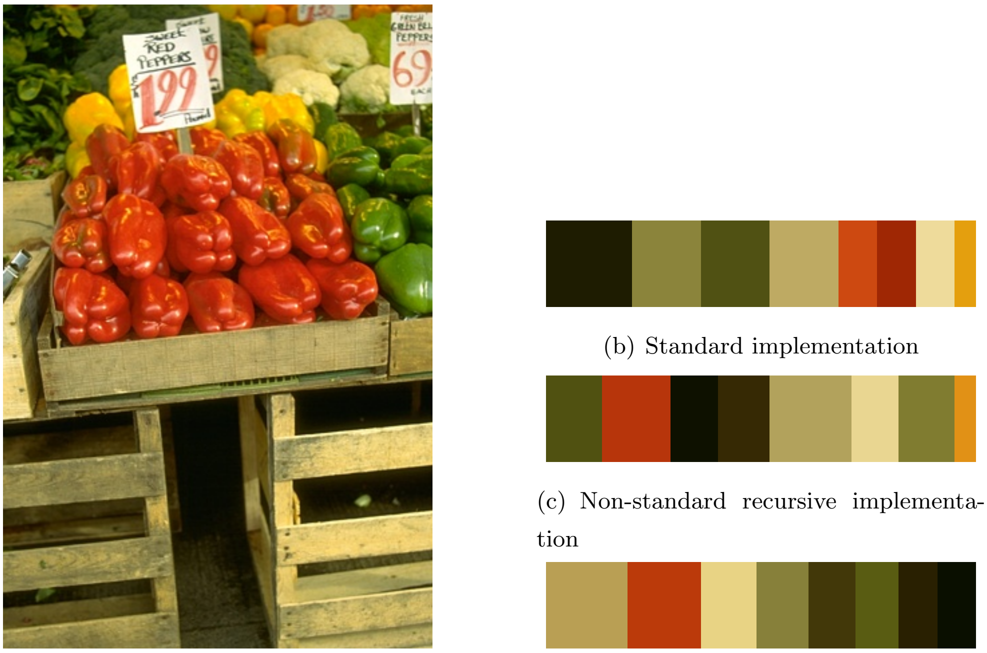
In the splitting cluster step, the cluster with the highest distortion represented by the centroid $\mathbf{c}_i$ is divided into two new clusters by adding a perturbation vector ϵ. The distortion D_i in the ith cluster C_i is given by:

$$D_i = \sum_n h(n)||\mathbf{x}(n) - \mathbf{c}_i||^2, \qquad \mathbf{x}(n) \in C_i \tag{3.2}$$

where $\mathbf{c}_i$ is the centroid of cluster C_i, $\mathbf{x}(n)$ is the color vector at pixel n and $h(n)$ is the perceptual weight for pixel n. $h(n)$ has been set to 1 for all pixels in the implementation so as to reduce its computation load. In order to obtain the perturbation vector, the direction of maximum variance in the color space is computed. This direction is given by the eigenvector corresponding to the maximum eigenvalue of the covariance matrix. Then, the deviation in this direction is computed and is assigned to each component of the perturbation vector. Consequently, the two new centroids are obtained as $\mathbf{c}_i + \epsilon$ and $\mathbf{c}_i - \epsilon$.

In the updating centroids step, the Generalized Lloyd Algorithm [GG93] is used. This algorithm calculates the centroids of each cluster and constructs a new partition by associating each point with the closest centroid. Then the centroids are recomputed for the new clusters and the algorithm is repeated by alternate application of these two steps until the distortion change is lower than a given percentage (in the MPEG-7 recommendation this parameter is set to 5%) or the maximum number of clusters is reached (as previously commented, MPEG-7 recommends eight). Figure 3.1(b) depicts the dominant color values that have been extracted from Figure 3.1(a) using the technique specified in the MPEG-7 standard.

The second implemented technique, which is not specified in the standard, is only based on the Generalized Lloyd Algorithm. Instead of beginning with only one cluster, this algorithm starts with eight centroids assigned uniformly in the color space. Then, each centroid is recalculated and each point is assigned to the closest centroid until a minimum distortion is met. This implementation has some advantages, such as its low complexity and, as a consequence, its low computing time. The main drawback is that the dominant colors resulting of this implementation could be biased because of the initial assignment of the centroids. However, this problem is overcome since this implementation allows computing the descriptor recursively, so the eight most dominant colors of the two children in the BPT, i.e. the eight dominant color values which represent the eight highest number of pixels, are assigned to the initial centroids of their father instead of being uniformly assigned. The dominant color values obtained from Figure 3.1(a) using recursive and non-recursive non-standard implementations are shown in Figures 3.1(c) and 3.1(d), respectively. The results obtained by the



(a) Image used for Dominant Color Descriptor extraction

(b) Standard implementation

(c) Non-standard recursive implementation

(d) Non-standard non-recursive implementation

Figure 3.1: Dominant Color Descriptor extraction from the image shown in (a) using various implemented techniques (b) (c) (d)

recursive implementation are more similar to the MPEG-7 technique implementation results than the results obtained by the non-recursive one are. That is the reason why, the proposed technique will be only used with the recursive implementation.

Once the centroids have been obtained by any of these methods, the following steps are common for both techniques. The centroids which are not at least T_d distance apart are merged forming only one centroid. Then, the percentage of each dominant color p_i is computed as the fraction of pixels in the image or region belonging to each centroid. The optional color variance v_i is computed for each dimension of the color space and for each dominant color. Finally, the spatial coherency s is obtained.

Both techniques are performed in the YUV color space as it is recommended in [MPS02], so the first step is to convert the image or image region values to this color space. This conversion is done by using the YUV Color Descriptor, which is presented in Section 3.2.4. However, the resulting dominant color values are given in the RGB color space according to MPEG-7. Furthermore, both implemented techniques work with the color histogram computed in the YUV color space instead of working with the image or image region directly, allowing the descriptor to be computed recursively. This histogram is obtained by using the Color Histogram Descriptor, which is also presented in Section 3.2.4.

3.2.2 Color Structure Descriptor

The *Color Structure Descriptor* (CSD) represents an image by both the color distribution of the image or image region (similar to a color histogram) and the local spatial structure of the color. The extra spatial information makes the descriptor sensitive to certain image features to which an ordinary color histogram is blind. For example, the results of applying this descriptor for the images in Figure 3.2 will be different while the results obtained by the color histogram would be the same.

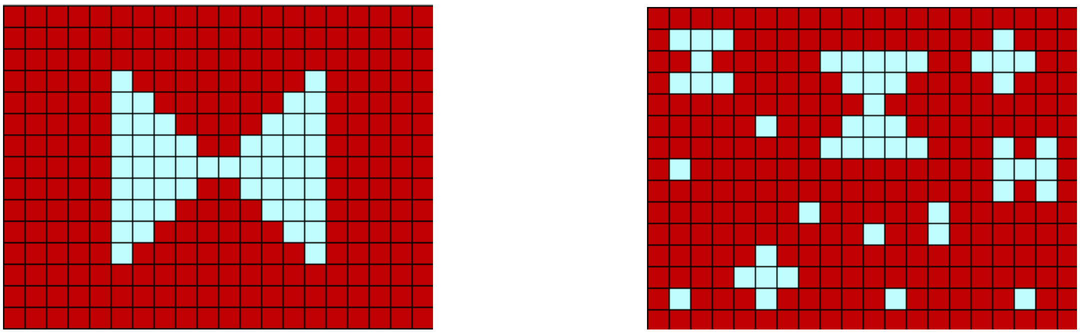
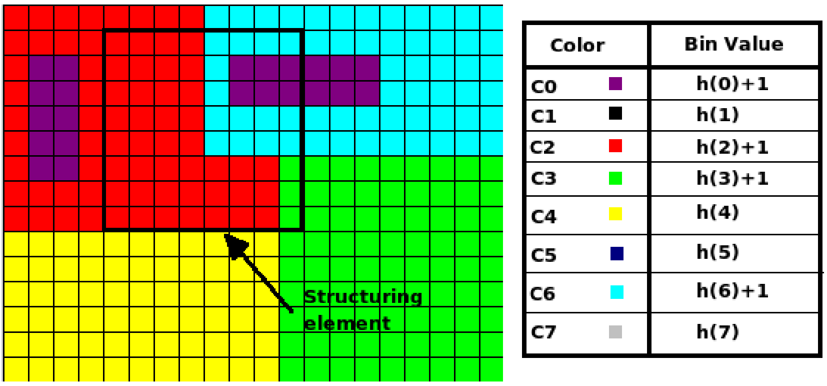


Figure 3.2: Two images with different local spatial structure of the color but with the same color histogram

In order to express local color structure, a structuring element is used. This descriptor counts the number of times a particular color is contained within the structuring element while the image or image region is scanned by this structuring element. Therefore, the descriptor consists on a color structure histogram than can be denoted by $\bar{h}_s(m)$, $m \in \{1, ..., M\}$, where the value in each bin represents the number of structuring elements containing one or more pixels with color c_m. Before scanning the image, a non-uniformly quantization of the colors in the HMMD color space in M cells is performed, where M is chosen from the set 256, 128, 64, 32. The scale of the associated square structuring element is denoted by s, which is commonly set to 8.



Color		Bin Value
C0	🟪	h(0)+1
C1	⬛	h(1)
C2	🟥	h(2)+1
C3	🟩	h(3)+1
C4	🟨	h(4)
C5	🟦	h(5)
C6	🟦	h(6)+1
C7	⬜	h(7)

Figure 3.3: Accumulation of color structure histogram

Related to the implementation, the user can choose both the size of the structuring element s and the number of cells M in which the HMMD color space is divided. Firstly, a subsampling factor [MPS02] is computed because the standard calls for images that deviate from a nominal size to be uniformly subsampled. Next, the image representation in the M cell-quantized HMMD color space is obtained. Finally, the structuring element scans the image such that the element visits every position in the pixel grid and the element always lies entirely within the image. For each position, if a quantized color c_m belongs to the element, the corresponding bin of the histogram is incremented in one unit, independently of the number of pixels which have the same value c_m. In Figure 3.3, an illustrative example with a quantization of 8 levels is shown. In the current position visited by the structuring element, the bins of the structure color histogram corresponding to the four quantized colors belonging to the element are incremented.

For image regions, the extraction of the Color Structure Descriptor is specified in [MP8]. The pixels that form the arbitrarily shaped region are called *active pixels*, whereas the pixels that do not belong to the region are known as *passive pixels* (see Figure 3.4). First, the bounding box of the image region is computed in order to work with a rectangular shape. Then, the structuring element visits every position in the pixel grid in which it lies entirely within the bounding box. Now, for each position, the passive pixels that are within the structuring element are not taken into account. Thus, only active pixels participate in the extraction of the descriptor.

Since the Color Structure Descriptor consists of a histogram that depends on the local spatial structure of the color, no recursive implementation is feasible here.

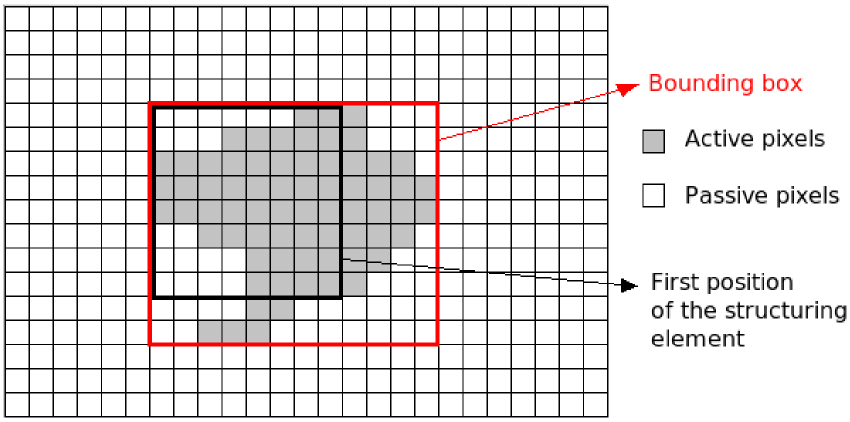


Figure 3.4: Color Structure Descriptor extraction for image regions

3.2.3 Color Layout Descriptor

The *Color Layout Descriptor* captures the spatial layout of the representative colors on a grid superimposed on a region or image. The representation is based on coefficients of the Discrete Cosine Transform.

The extraction process of the descriptor from an image [MPS02] consists of four stages: image partitioning, representative color detection, DCT transformation, and nonlinear quantization of the zigzag-scanned coefficients (see Figure 3.5). In the first stage, the image is divided into 64 blocks (8 blocks x 8 blocks). Since the sizes of the input image are not necessarily multiple of 8, it is assumed that the blocks can differ in their size, although the pixels are distributed in the most uniform way. In the following stage, a single representative color is selected from each block. Consequently, a tiny image representation of size 8x8 is obtained. Any method to compute each representative color can be applied, but the average of the pixel colors in a block is sufficient in general. In the third stage, each of the three color components is transformed by a 8x8 DCT, so three sets of 64 DCT coefficients are obtained. It is recommended by the standard to use the YUV color space. In the last stage, each set of coefficients is zigzag-scanned and a few low-frequency coefficients are nonlinearly quantized. In the MPEG-7 standard, it is recommended to use a total of 12 coefficients, 6 for luminance and 3 for each chrominance. Nevertheless, the implementation has been performed in order to allow the user to choose how many coefficients wants to use for each component.

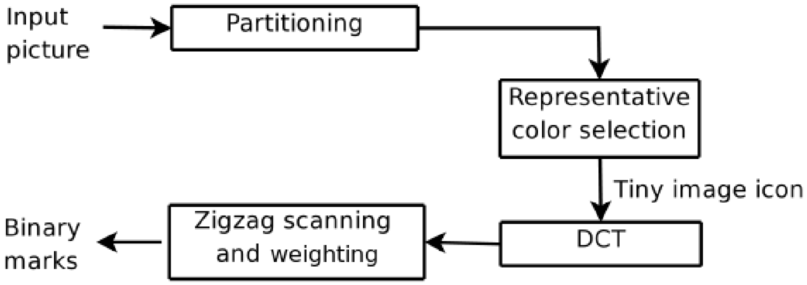


Figure 3.5: Extraction process of Color Layout Descriptor

Some assumptions had to be made in order to implement this descriptor for regions, because it is not specified in the standard. As a consequence of that, the bounding box descriptor is computed in order to divide the region into 64 blocks. In the second stage, the blocks which lie entirely within the region are treated in the same way as in the image case, i.e. the average of the pixel colors in these blocks are computed directly. On the other hand, the boundary blocks, i.e. the blocks which do not lie entirely within the region but contain some of its pixels, are padded before computing their average and the empty blocks, i.e. the blocks which do not contain any pixel of the region, are not considered. In the third stage, if there are not empty blocks, the 2-D DCT is applied in the same way as in the image case, whereas if there are some empty blocks, the shape-adaptive DCT must be computed [Sik95]. The shape-adaptive DCT consists of four steps. In the first step, a vertical shift is applied. Next, the 1-D DCT has to be computed for each column of samples. In the third step, a horizontal shift is applied so all the samples are placed together in the top-left corner of the 8x8 image. Eventually, the 1-D DCT is computed for each row. Once this transform has been applied, the resulting DCT coefficients are zigzag-scanned and the descriptor is obtained.

No recursive implementation is feasible for this descriptor. Therefore, we cannot take advantage of BPT structure for its computation.

3.2.4 Auxiliary Color Descriptors

Some color descriptors, which are not specified in the MPEG-7 standard, have been implemented in order to assist the descriptors that have already been presented in this section. All of them are briefly introduced below:

- The *Color Mean Descriptor* computes the average of all pixels belonging to the image or region in each dimension, so a representative color value is obtained. It is computed in the color space in which the image or region has been given. Its computation is done in a recursive way.

- The *Color Variance Descriptor* computes the variance of color values of an image or region in each dimension. In its implementation, the Color Mean Descriptor has to be previously obtained in order to calculate the color variance. The Color Variance Descriptor has been implemented in order to be recursively computed.

- The *Color Histogram Descriptor* allows a representation of the distribution of colors in an image or region, derived by counting the number of pixels of each given set of color ranges in a color space. In other words, a histogram of an image is produced first by discretization of the colors in the image into a number of bins, and second by counting the number of image pixels in each bin. This descriptor has a recursive implementation.

- The *YUV Color Descriptor* converts the color values of an image or region, which has been given in the RGB color space, into the YUV color space, i.e. the image or region is obtained in this new color space. No recursive implementation is feasible for this descriptor.

- The *Gray Color Descriptor* converts the color values of an image or region, which has been given in the RGB or YUV color space, into the Monochrominance color space, i.e. the image or region is obtained in the gray scale. No recursive implementation is possible here.

3.3 Texture Descriptors

Image texture is an important visual attribute for searching and browsing through large collections of similar looking patterns [MPS02]. Although it is easy to understand what one means by texture, there is no universally accepted formal definition of texture. This section provides the reader with an overview of a selection of descriptors that are considered by the MPEG-7 group for describing visual content based on texture.

3.3.1 Texture Edge Histogram Descriptor

The *Texture Edge Histogram Descriptor* (EHD) captures the spatial distribution of edges. A given image is first sub-divided into 4 x 4 subimages, and local edge histograms for each of these sub-images are computed. In order to generate the histogram, edges are categorized into five types: vertical, horizontal, 45° diagonal, 135° diagonal, and non-directional edges. Thus, each local histogram has five bins corresponding to the above five categories. The image partioned into 16 sub-images results in 80 bins. Table 3.1 summarizes the semantics of each bin.

Related to the implementation, the five categories of edges can be extracted by a block-based edge extraction scheme. In order to do that, each subimage is subdivided into nonoverlapping image blocks. The size of the image block depends on the image resolution because the number of image blocks per subimage is kept constant, independently of the original image dimensions. As cited in [PJW00], experiments show that a number of image blocks around 1100 seems to capture good directional

Table 3.1: Semantics of the histogram bins of the EHD

H_E	Semantics
h(0)	Relative population of vertical edges in subimage at (0,0)
h(1)	Relative population of horizontal edges in subimage at (0,0)
h(2)	Relative population of 45° edges in subimage at (0,0)
h(3)	Relative population of 135° edges in subimage at (0,0)
h(4)	Relative population of non-directional edges in subimage at (0,0)
⋮	⋮
h(75)	Relative population of vertical edges in subimage at (3,3)
h(76)	Relative population of horizontal edges in subimage at (3,3)
h(77)	Relative population of 45° edges in subimage at (3,3)
h(78)	Relative population of 135° edges in subimage at (3,3)
h(79)	Relative population of non-directional edges in subimage at (3,3)

edge features. Each image block is then partioned into 2 x 2 block of pixels. Figure 3.6 depicts the whole partitioning process. The edge detector operators (see Figure 3.7) are then applied to these 2 x 2 blocks, treating each block as a pixel and the average intensity as the corresponding block intensity value. Then, the edge detector with the maximum edge strength is identified. If this edge strength is above a given threshold, then the corresponding edge orientation is associated with the image block. If the maximum of the edge strengths is below the given threshold, then that image block is not classified as an edge block. For example, a homogenous image block would not be classified as an edge block. Since there are image blocks without any edge, the sum of the five normalized histogram bins for each subimage could be less than 1.

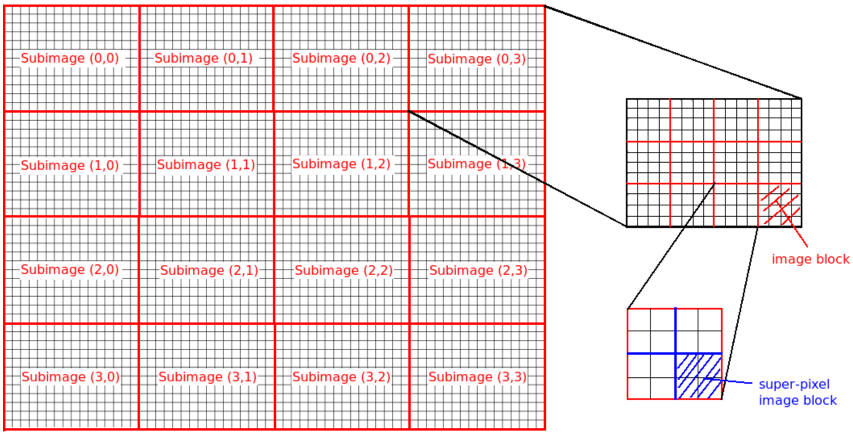


Figure 3.6: Texture Edge Histogram Descriptor partitioning process

$$\begin{array}{|c|c|}\hline 1 & -1 \\ \hline 1 & -1 \\ \hline \end{array}$$

(a) vertical

$$\begin{array}{|c|c|}\hline 1 & 1 \\ \hline -1 & -1 \\ \hline \end{array}$$

(b) horizontal

$$\begin{array}{|c|c|}\hline \sqrt{2} & 0 \\ \hline 0 & -\sqrt{2} \\ \hline \end{array}$$

(c) 45°

$$\begin{array}{|c|c|}\hline 0 & \sqrt{2} \\ \hline -\sqrt{2} & 0 \\ \hline \end{array}$$

(d) 135°

$$\begin{array}{|c|c|}\hline 2 & -2 \\ \hline -2 & 2 \\ \hline \end{array}$$

(e) non- directional

Figure 3.7: Edge detector operators

Some assumptions had to be made in order to implement this descriptor for regions, because it is not specified in the standard. As a consequence of that, in the first place, the bounding box descriptor is computed in order to divide the region into 4 × 4 subimages. Then, each subimage is partioned into image blocks. Those ones which do not lie entirely within the region will not be taken into account. Thus, the edge detector operators are only applied to the image blocks lying completely inside the region in order to extract the EHD.

Since this descriptor contains local spatial information about the edges, no recursive implementation is feasible here.

3.3.2 2D-Discrete Wavelet Transform

The *2D-Discrete Wavelet Transform* (2D-DWT) allows the decomposition of an image in various frequency subbands. It consists in applying a high-pass and a low-pass filters to the original signal rows in the first step. Then, the samples of each row are downsampled by 2, so the number of columns decreases to its half. Then, the columns of the signal resulting from the high-pass filter are filtered by both of them and their columns are downsampled obtaining as a result the signal containing the high frequencies in the XY direction (High-High subband) which includes the diagonal details, and the signal containing the high frequencies in the X direction (High-Low subband),which represents the vertical details. In the same way, both filters are also applied to the columns of signal obtained from the first low-pass filter and the columns of the resulting images are downsampled by 2, i.e. the number of rows is divided by 2. As a consequence, the signals containing the high frequencies in the Y direction (Low-High subband) representing the horizontal details, and the low frequencies, which is known as the approximation image, are extracted. These four images are the result of one level of the decomposition. Then, each new level is obtained by repeating the same process on the approximation image. Figure 3.8 shows the scheme of each decomposition step.

There are different families of filters which can be used for 2D-DWT such as Haar, Daubechies, etc. Related to the implementation, all these kinds of filters can be easily added by the user in order to get the decomposition with the desired ones. Specifically, the *Haar Wavelet Descriptor* has been implemented as a particular case of 2D-DWT in which the low-pass Lo and the high-pass Hi filters of the decomposition are the Haar filters:

$Lo = [1, 1]$ and $Hi = [-1, 1]$

An example of the result of applying the Haar Wavelet Descriptor with one level of

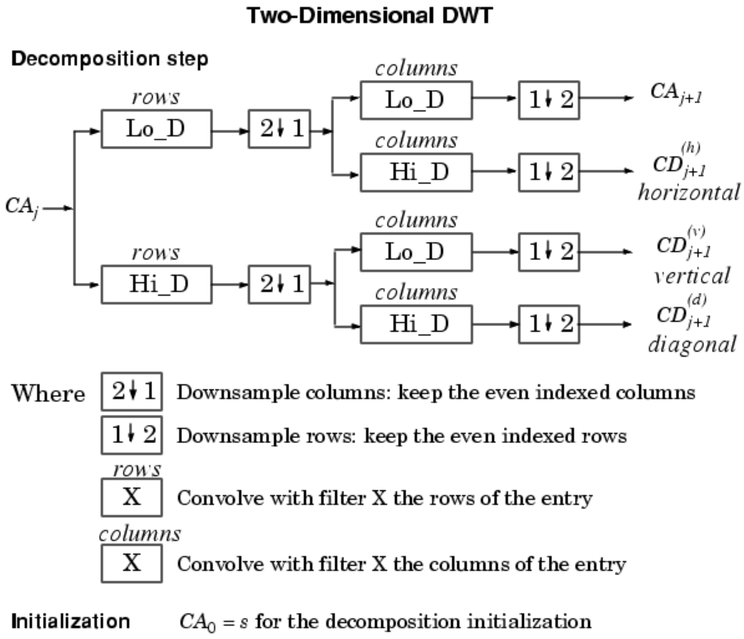

Two-Dimensional DWT
Decomposition step
rows
Lo_D
2↓1
columns
Lo_D
1↓2
CA_{j+1}
columns
Hi_D
1↓2
$CD^{(h)}_{j+1}$
horizontal
CA_j
rows
Hi_D
2↓1
columns
Lo_D
1↓2
$CD^{(v)}_{j+1}$
vertical
columns
Hi_D
1↓2
$CD^{(d)}_{j+1}$
diagonal
Where
2↓1 Downsample columns: keep the even indexed columns
1↓2 Downsample rows: keep the even indexed rows
rows
X Convolve with filter X the rows of the entry
columns
X Convolve with filter X the columns of the entry
Initialization $CA_0 = s$ for the decomposition initialization

Figure 3.8: 2D-DWT Decomposition step

decomposition is shown in Figure 3.9 where the top-left image is the approximation, the top-right one contains the horizontal details, the bottom-left image includes the vertical details, and the bottom-right one is the image containing the diagonal details.

3.3.3 Haar-Power Descriptor

The *Haar-Power Descriptor* is a descriptor that is not specified in the MPEG-7 standard. It represents the energies of the Haar wavelet coefficients associated to the region pixels for the Low-High (LH), High-Low (HL), and High-High (HH) subbands in each level of the decomposition. These subbands result of the Haar Wavelet Descriptor (see Section 3.3.2). Once the original signal has been decomposed, the energy for each subband and in each level is computed.

This descriptor can be computed either on images or regions, although for regions is necessary that the 2D-DWT (see Section 3.3.2) has been computed for the whole image where the region belongs to. Due to the simplicity of the energy calculation, the Haar-Power descriptor can be computed recursively for each subband. As a

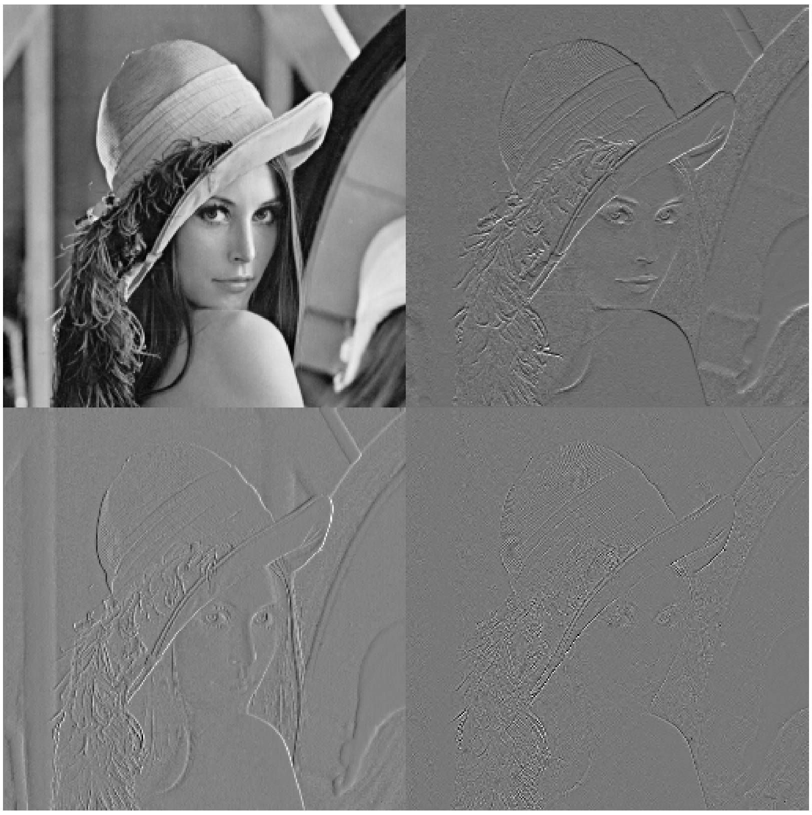


Figure 3.9: 2D-DWT Decomposition with Haar Wavelet Descriptor

consequence, the energy for a certain subband and in a certain level for a father node is obtained by adding the energies of its children for this subband and this level.

3.4 Shape Descriptors

Object shape features provide a powerful clue to object identity and functionality, and can even be used for object recognition. Humans can recognize characteristic objects solely from their shapes which proofs that shape often carries semantic information. This distinguishes shape from other elementary visual features, such as color, motion, or texture, which, while equally important, usually do not reveal object identity.

Two descriptors characterize the different shape features of a 2D object or region. The Region Shape descriptor captures the distribution of all pixels within a region. The Contour Shape descriptor characterizes the shape properties of the contour of an object. The curvature scale space (CSS) technique was selected as a contour shape descriptor for MPEG-7.

3.4.1 Contour-Shape Descriptor

The *Contour-Shape Descriptor* [MP3] is based on the Curvature Scale Space (CSS) representation of the contour. In order to create a CSS description of a contour shape, *N_samples* equidistant points have to be selected on the contour, starting from an arbitrary point and following the contour clockwise. The x-coordinates of the selected points are grouped together and the y-coordinates are also grouped together into two series x, y. As a consequence, an arc-length parametrization $r(u) = (x(u), y(u))$ is obtained. The CSS representation is computed by convolving this parametric representation of the curve with a Gaussian function, as the standard deviation of the Gaussian varies from a small to a large value, and extracting the curvature zero-crossing points of the resulting curves [MAK96]. As a result of the smoothing, the contour evolves and its concave parts gradually flatten-out, until the contour becomes convex. So, the CSS representation decomposes the contour into convex and concave sections by determining the inflection points (e.g. points at which curvature is zero). This representation is essentially invariant under rotation, uniform scaling, and translation of the curve.

A so-called CSS image can be associated with the contour evolution process. The CSS image horizontal coordinates correspond to the indices of the contour points selected to represent the image, and CSS image vertical coordinates correspond to the amount of filtering applied, defined as the number of passes of the filter. So, the curvature zero-crossing points resulting of each smoothing step are represented in the CSS image. The coordinate values of the prominent peaks in the CSS image are extracted and ordered based on decreasing values of vertical coordinates (amount of smoothing). Figure 3.10 shows an example of CSS representation obtained by this descriptor. In addition, the eccentricity and circularity of both the smoothed contour and the original contour are also computed.

As a consequence, the descriptor consists of the eccentricity and circularity values of the original and filtered contour, the index indicating the number of peaks in the CSS image, the magnitude (amount of smoothing) of the largest peak and the x-positions,

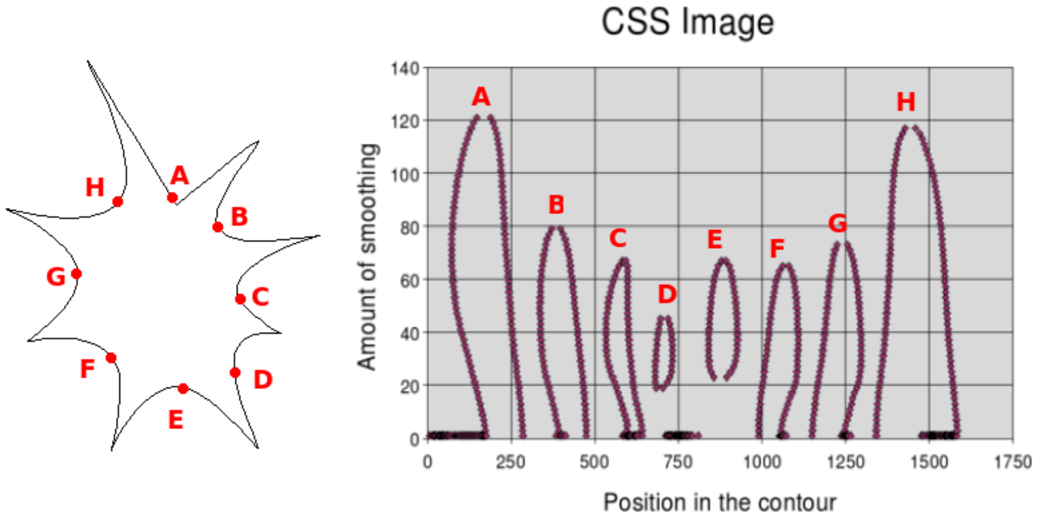


Figure 3.10: CSS Image representation

y-positions and height on the remaining peaks (positions are relative to the highest peak and height is related to the previous peak's height).

Next, each step in the extraction process of this descriptor will be explained in detail. Once the arc-length parametrization of the contour is obtained, the circularity and the eccentricity parameters of the original contour are computed as follows:

$$circularity = \frac{perimeter^2}{area} \tag{3.3}$$

$$eccentricity = \sqrt{\frac{i_{20} + i_{02} + \sqrt{i_{20}^2 + i_{02}^2 - 2\, i_{20}\, i_{02} + 4\, i_{11}^2}}{i_{20} + i_{02} - \sqrt{i_{20}^2 + i_{02}^2 - 2\, i_{20}\, i_{02} + 4\, i_{11}^2}}} \tag{3.4}$$

where

$$i_{02} = \sum_{k=1}^{N} (y_k - y_c)^2, \quad i_{20} = \sum_{k=1}^{N} (x_k - x_c)^2, \quad i_{11} = \sum_{k=1}^{N} (x_k - x_c)(y_k - y_c),$$

N is the number of contour samples, (x_k, y_k) are the coordinates of each contour point, and (x_c, y_c) is the center of mass of the contour shape. These two parameters are included in a vector which is called $GlobalCurvature$.

Then, the calculation of the curvature function is as follows:

$$K(u,\sigma) = \frac{X_u(u,\sigma)Y_{uu}(u,\sigma) - X_{uu}(u,\sigma)Y_u(u,\sigma)}{(X_u(u,\sigma)^2 + Y_u(u,\sigma)^2)^{3/2}} \tag{3.5}$$

where

$$X(u,\sigma) = x(u) * g(u,\sigma) \quad , \qquad Y(u,\sigma) = y(u) * g(u,\sigma)$$

$$X_u(u,\sigma) = x(u) * g_u(u,\sigma) \quad , \qquad X_{uu}(u,\sigma) = x(u) * g_{uu}(u,\sigma)$$

$$Y_u(u,\sigma) = y(u) * g_u(u,\sigma) \quad , \qquad Y_{uu}(u,\sigma) = y(u) * g_{uu}(u,\sigma)$$

and $g(u,\sigma)$ is a 1-D Gaussian Kernel of width σ. According to the properties of convolution, the derivates of $X(u,\sigma)$ and $Y(u,\sigma)$ can be easily obtained by convolutioning each signal with the derivate of the well-known $g(u,\sigma)$.

First, this function is computed using some values of σ until it has no zero-crossing points in order to estimate the highest peak and decide which increment of σ has to be used for each smoothing step. For example, it would has no sense to increment σ in 0.1 steps if the highest peak was found when $\sigma = 100$. This decision has been taken so that the computational cost is lower.

Then, for each smoothing step, which is result of incrementing σ with the value previously obtained, the curvature function is calculated and its zero-crossing points obtained. These points are stored with their respective value of σ in order to create the CSS representation in the end. This process is carried out until no zero-crossing points are found. Furthermore, the result of the convolution has been decided to be downsampled for high values of σ in order to reduce computational cost because the number of samples of the gaussian kernel is proportional to sigma. In this way, this decision does not affect the result because the samples are more concentrated where the sign of curvature function changes.

Once the final filtered contour has been obtained, its circularity and eccentricity parameters are computed according to Equations 3.3 and 3.4, respectively. In order to calculate them, an approximation of the area is obtained by using the Riemann integration method while the perimeter is computed as sum of the distances between the samples. These two parameters are represented by a vector which is called $PrototypeCurvature$.

Eventually, the localization and the heigth of the CSS Image's peaks are extracted. This process consists of two steps. The first one assigns zero-crossing points to the peaks already detected. Therefore, the pair of zero-crossing points of the curvature function nearer each peak for the current σ are searched and assigned to it. In the second step, the remaining points are grouped two by two in order to extract the new peaks. Next, the σ is decreased and these two steps are repeated until the initial value for σ is reached.

This descriptor has to be computed independently for each region. Thus, no recursive implementation is feasible here.

3.4.2 Auxiliary Shape Descriptors

Some shape and geometric descriptors, which are not specified in the MPEG-7 standard, have been implemented in order to assist the descriptor that has already been presented. All of them are briefly introduced below:

- The *Area Descriptor* computes the number of pixels that belongs to the region. It is computed recursively.

- The *Bounding Box Descriptor* represents the horizontal circumscribed rectangle of the region. It contains the coordinates corresponding to the top-left and bottom-right vertices. It is computed for a regions and its computation is recursive.

- The *Localization Descriptor* represents the Bounding Box Descriptor as well as the region center of mass. It is computed recursively.

- The *Oriented Bounding Box Descriptor* represents the circumscribed rectangle of the region that is computed according to their principal axes. It contains the coordinates of the four vertices. No recursive implementation is feasible for this descriptor.

- The *Perimeter Descriptor* computes the perimeter of a region, i.e. the number of pixel's edges that constitute the contour. This descriptor is computed recursively.

- The *Mask Descriptor* is an image with boolean values in which the pixels corresponding to the region are set to true. Its computation is not recursive.
- The *ContourMask Descriptor* is an image with boolean values in which the pixels corresponding to the contour's region are set to true. No recursive implementation is feasible for this descriptor.

Chapter 4

Similarity Matching

4.1 Introduction

The aim of this chapter is to present the various distances implemented for each descriptor. These distances are useful for image retrieval because they give a grade of similarity between two descriptors of the same type. Thus, if a query image is given, the distance between its descriptors and the rest of images' descriptors are computed and, then, the images which are more similar to the query image are obtained. In this chapter, each descriptor will be considered independently of the others. In other words, descriptors will be assessed isolately. First, the distances implemented for color descriptors are presented in Section 4.2. Next, similarity matching for texture descriptors is analyzed in Section 4.3. Shape descriptor distances will not be studied because they are useless for image retrieval, although they will be very useful for region retrieval. In this chapter, each similarity measure will be studied independently, i.e., each descriptor will be analyzed individually. In the next chapter, the fusion of different descriptor distances will be studied in order to take into account the combination of the whole set of descriptors computed.

All the images used in this chapter have been extracted from a Corel dataset in order to give some illustrative examples. We have used a set of 1000 images from this database instead of using the whole given dataset. In Chapter 6, where the experimental results are presented, we have given more importance to which database has to be considered. For this reason, the same database used by MPEG-7 to obtain their results has been chosen. This dataset, which is called Common Color Dataset (CCD), is also introduced in Chapter 6.

4.2 Color Descriptors Similarity Matching

In this section, the distances implemented for color descriptors are presented. Some of them are recommended by MPEG-7, but other distances, which are not included in the standard, will be also analyzed.

4.2.1 Dominant Color Similarity Matching

First, the distance between two Dominant Color descriptors proposed in the MPEG-7 standard is analyzed. Consider two DCDs,

$$\begin{aligned} F_1 &= \{(\mathbf{c}_{1i}, p_{1i}, v_{1i}), s_1\}, \qquad (i = 1, 2, ..., N_1) \text{ and} \\ F_2 &= \{(\mathbf{c}_{2j}, p_{2j}, v_{2j}), s_2\}, \qquad (j = 1, 2, ..., N_2) \end{aligned}$$

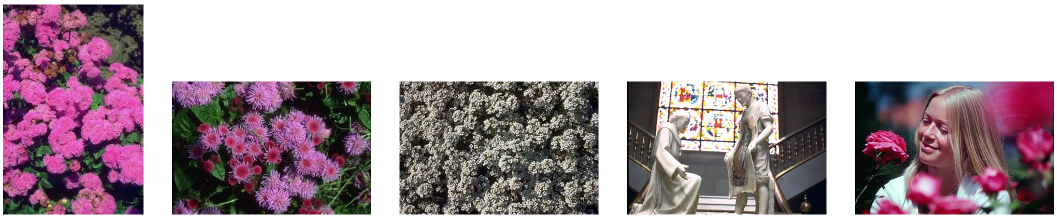
Ignoring the optional variance parameter and the spatial coherence, the dissimilarity $D(F_1, F_2)$ between the two descriptors can be computed [MPS02] as:

$$D^2(F_1, F_2) = \sum_{i=1}^{N_1} p_{1i}^2 + \sum_{j=1}^{N_2} p_{2j}^2 - \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} 2\, a_{1i,2j}\, p_{1i}\, p_{2j} \tag{4.1}$$

where the subscripts 1 and 2 in all variables stand for descriptors F_1 and F_2, respectively, and $a_{k,l}$ is the similarity coefficient between two colors c_k and c_l,

$$a_{k,l} = \begin{cases} 1 - d_{k,l}/d_{max} & d_{k,l} \leq T_d \\ 0 & d_{k,l} > T_d \end{cases}$$

where $d_{k,l} = ||c_k - c_l||$ is the Euclidean distance between the two colors c_k and c_l, T_d is the maximum distance for two colors to be considered similar, and $d_{max} = \alpha T_d \quad (\alpha \in \mathbb{R}, \quad \alpha \geq 1)$. In particular, this means that any two dominant colors from one single



(a) query (b) retrieve #1 (c) retrieve #2 (d) retrieve #3 (e) retrieve #4

Figure 4.1: Results of query with MPEG-7 dissimilarity measure ignoring optional variance parameter and the spatial coherence

description are at least T_d distance apart. A recommended value for T_d is between 10 to 20 in the YUV color space and for α is between 1.0 to 1.5. The maximum value of the distance given in Equation 4.1 is 2.0, which is reached when all the coefficients $a_{k,l}$ are 0 and each of the other summations is 1.0.

Regarding the implementation, the threshold T_d^2 is set to 255.0 because this value is recommended in [MP8] and the parameter α is set to 1.0. One of the problems detected about this measure is that the distance between the query image and an image that does not match any dominant color can be smaller than the distance to another one with one matching at least. This happens because when the parameters $a_{k,l}$ from the dissimilarity measure in Equation 4.1 are almost 0, the nearest image to the query image is considered to be the most homogeneous one, i.e., the image that has the percentages of its dominant color values most similar. This fact is not important when the database is big enough because the probability of no matchings between dominant color values is small. However, a change has been made in order to tackle this problem, so when no matchings are found between two Dominant Color descriptors, the distance between these descriptors is set to 2.0, i.e., the maximum value of the dissimilarity measure. This adjustment allows the user working with smaller data bases. Some results obtained with this distance are shown in Figure 4.1. This figure gives an illustrative example of the images retrieved when we are looking for pink flowers (Figure 4.1(a)). In this example, we can observe than dark colors, i.e. colors that have a small value for luminance, can have a negative influence. This happens for images given by Figures 4.1(c) and 4.1(d), in which the darkest colors have been matched.

One variation of the above distance proposed by MPEG-7 is to use the spatial coherence field [MPS02]. In the MPEG-7 experiments, the following distance was used:

$$D_s = (w_1 \left| s_1 - s_2 \right| + w_2)D \tag{4.2}$$

where s_1 and s_2 are the spatial coherencies of the query and target descriptors and w_1 and w_2 are fixed weights, with recommended settings to 0.3 and 0.7, respectively.

Another variation proposed by MPEG-7 consists on taking into account the optional variance parameter. According to [MPS02], if the color variance field is present, the matching function is based on modeling the color distribution as a mixture of Gaussian distributions with parameters defined as color values and color variance. Calculation of the squared difference between the query and target distributions then leads to the following formula for the matching function:

$$D_v = \sum_{i=1}^{N_1}\sum_{j=1}^{N_1} p_{1i}\, p_{ij}\, f_{1i\,1j} + \sum_{i=1}^{N_2}\sum_{j=1}^{N_2} p_{2i}\, p_{2j}\, f_{2i\,2j} - \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} 2\, p_{1i}\, p_{2j}\, f_{1i\,2j} \tag{4.3}$$

where

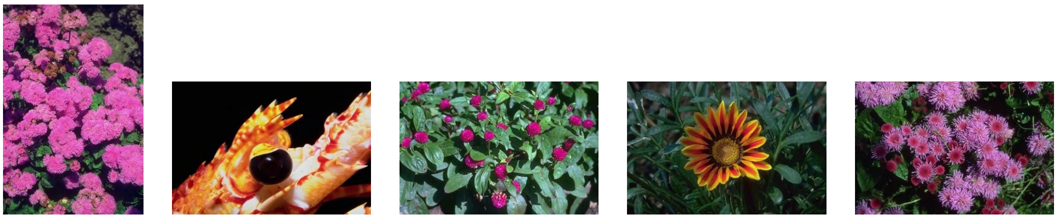
$$f_{x_i\,y_j} = \frac{1}{2\pi\sqrt{{v_{x_i\,y_j}}^{(l)}{v_{x_i\,y_j}}^{(u)}{v_{x_i\,y_j}}^{(v)}}} exp\left[-\left(\frac{{c_{x_i\,y_j}}^{(l)}}{{v_{x_i\,y_j}}^{(l)}} + \frac{{c_{x_i\,y_j}}^{(u)}}{{v_{x_i\,y_j}}^{(u)}} + \frac{{c_{x_i\,y_j}}^{(v)}}{{v_{x_i\,y_j}}^{(v)}}\right)/2\right]$$

and

$${c_{x_i\,y_j}}^{(l)} = ({c_{x_i}}^{(l)} - {c_{y_j}}^{(l)})^2, \quad {v_{x_i\,y_j}}^{(l)} = ({v_{x_i}}^{(l)} + {v_{y_j}}^{(l)})$$

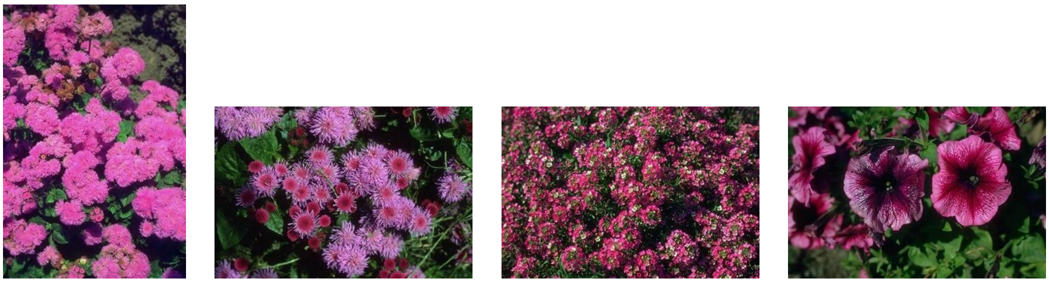
In the equations above, ${c_{x_i}}^{(l)}$ and ${v_{x_i}}^{(l)}$ are dominant color values and color variances, x, y index the query and target descriptors, i, j index the descriptor components and l, u and v index the components of the color space. The results obtained by taking account the color variance parameter are shown in Figure 4.2. In this case, it happens the same problem than in the previous figure. In two retrieved images, Figures 4.2(b) and 4.2(d), the pink color, which could be consider the most important perceptually in the query, has not been matched. The reason is the same than before. Thus, some dark colors have been matched.

In order to improve the results obtained from the MPEG-7 dissimilarity measure, two variants of this distance have been implemented. The first one is based on ensuring that a minimum percentage of pixels of the pair of images (query and target) belonged



(a) query (b) retrieve #1 (c) retrieve #2 (d) retrieve #3 (e) retrieve #4

Figure 4.2: Results of query with MPEG-7 dissimilarity measure considering color variance parameter

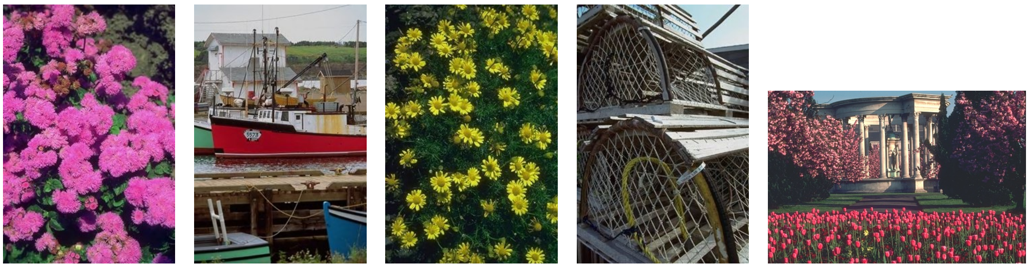


(a) query (b) retrieve #1 (c) retrieve #2 (d) retrieve #3

Figure 4.3: Results of query with a dissimilarity measure based on a minimum percentage set to 50%

to the dominant color values that have been matched. In other words, a minimum portion of the image has to be represented by the dominant color values that are less than T_d distance apart. So, the larger the minimum percentage is, the more accurate the descriptor becomes. The results obtained with a minimum percentage set to 50% are showed in Figure 4.3. As a consequence of this percentage, all the retrieved images (Figures 4.3(b), 4.3(c) and 4.3(d)) have pink color as one of the dominant color values.

The second variant is based on ensuring that a minimum number M of the N most dominant color values have been matched between them. For example, setting M to 4 and N to 5, the distances between two Dominant Color descriptors will be computed only if at least 4 of the 5 most dominant color values from one descriptor has been matched with 4 of the 5 most dominant color values from the other one. Consequently, this implementation ensures that the images resulting of the query image have in common a portion of the most representative color values. In Figure 4.4, the results obtained setting M to 4 and N to 5 are presented. From this example, this proposed



(a) query (b) retrieve #1 (c) retrieve #2 (d) retrieve #3 (e) retrieve #4

Figure 4.4: Results of query with a dissimilarity measure based on a minimum number M of the N most dominant color values have been matched, setting M to 4 and N to 5

dissimilarity measure seems not to have the expected behavior.

Another implementation for dissimilarity measure has been experimented. This is called *Merged Palette Histogram Similarity Measure* (MPHSM) [PW04] and is based on generating a common palette for the two Dominant Color descriptors by merging their color histogram bins using the threshold T_d. This is carried out in order to use the conventional histogram intersection similarity measure [RTG00]. In [PW04] the common palette is generated by searching the closest two colors between the two palettes and if this minimum distance is less than or equal to the threshold T_d then the two colors will be merged as:

$$c_{m(i,j)} = \frac{p_{1i}c_{1i} + p_{2j}c_{2j}}{p_{1i} + p_{2j}} \tag{4.4}$$

This process continues until the minimum distance is greater than the threshold T_d. The main drawback about this process is that it leads to a local optimization, so this is not the best way to select which colors have to be merged.

In order to obtain a global optimization, a variation has been included. Given two palettes $P_1 = \{c_{11}, ..., c_{1N_1}\}$ and $P_2 = \{c_{21}, ..., c_{2N_2}\}$, a bipartite graph that contains an edge between any pair of colors c_{1i} and c_{2j}, with its weight equal to the distance between them, is defined. Then, the objective is to find the maximum matching between the colors of P_1 and P_2 with the lowest weight. This problem is commonly known as *the assignment problem* [PM08] and can be efficiently solved using the Hungarian algorithm [Kuh55]. In particular, the weights of the edges between a pair of colors that are more than T_d distance apart are set to zero, and the rest of weights are decreased in T_d. This is made because the objective is not to obtain a maximum matching but to obtain the best matching between only the colors that are less than T_d distance apart. So, the colors from the palette P_1 that had been matched to colors from the palette P_2 by the Hungarian algorithm will be merged as the above Equation 4.4. The common palette $\{\{c_{mi}\}, i = 1, ..., N_m\}$ with N_m colors $(N_m \leq N_1 + N_2)$ is then generated with the use of the merged and unmerged colors from the two palettes. This merged palette forms a common color space for the two histograms, so the histograms of each Dominant Color descriptor, F_1 and F_2, can be redefined with use of this space. Consequently, the number of bins of each histogram is equal to N_m. Thus, the redefined F_1 and F_2 histograms are given by:

$$F_{1m} = \{\{c_{mi}, p_{1mi}\}, i = 1, ..., N_m\} \text{ and}$$
$$F_{2m} = \{\{c_{mi}, p_{2mi}\}, i = 1, ..., N_m\}$$

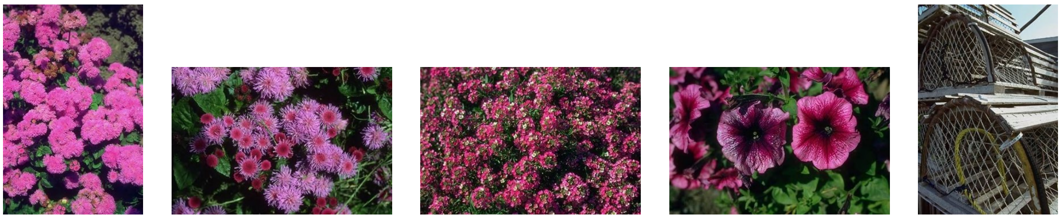
where $p_{xmi} = p_{xk}$ if the color c_{xk} from the palette P_x is the closest one to the color c_{xmi} from the merged palette. As these two histograms F_{1m} and F_{2m} are based on a common color palette, the histogram intersection method can be directly applied in order to compute their similarity. The MPHSM is defined as the intersection area of these two histograms, which is given by [RTG00]:

$$I(F_{1m}, F_{2m}) = \sum_{i=1}^{N_m} min(p_{1mi}, p_{2mi}) \tag{4.5}$$

The larger the value $I(F_{1m}, F_{2m})$ the more similar the two images, being the maximum value 1. Some results obtained by using Merged Palette Histogram Similarity Measure are shown in Figure 4.5. Except for Figure 4.5(e), the retrieved images can be considered appropiate because their dominant color values are rather similar to the query ones.

4.2.2 Color Structure Similarity Matching

The distance proposed by MPEG-7 for the Color Structure descriptor similarity matching is the L_1-norm [MvBE01], which is commonly used with other histogram



(a) query (b) retrieve #1 (c) retrieve #2 (d) retrieve #3 (e) retrieve #4

Figure 4.5: Results of query with Merged Palette Histogram Similarity Measure

descriptors. If two Color Structure descriptors are denoted by $h_{1s}(m)$ and $h_{2s}(m)$ with $m \in \{1, ..., M\}$, then the distance between them is computed by:

$$D(h_{1s}, h_{2s}) = \sum_{i=1}^{M} |h_{1s}(i) - h_{2s}(i)| \tag{4.6}$$

Other common similarity measures for histograms have been implemented [RTG00]. The first of them is the L_2-norm, commonly known as Euclidean norm, which is defined as:

$$D(h_{1s}, h_{2s}) = ||h_{1s} - h_{2s}||^2 = \sum_{i=1}^{M} (h_{1s}(i) - h_{2s}(i))^2 \tag{4.7}$$

The second one is the Histogram intersection which is given by:

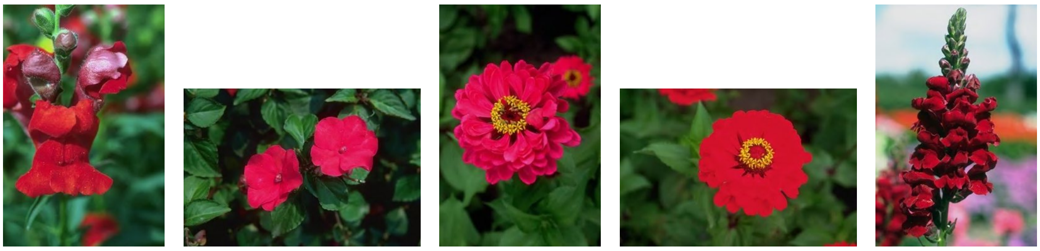
$$D(h_{1s}, h_{2s}) = 1 - \frac{\sum_{i=1}^{M} min(h_{1s}(i), h_{2s}(i))}{\sum_{i=1}^{M} h_{2s}(i)} \tag{4.8}$$

Eventually, the distance that is known as Jeffrey divergence has been implemented. This is a modification of the Kullback-Leibler divergence that is numerically stable, symmetric and robust with respect to noise and the size of histogram bins. Like the K-L divergence, from the information theory point of view, the Jeffrey distance has the property that it measures how inefficient on average it would be to code one histogram using the other as the code-book. It is defined as:

$$D_J(h_{1s}, h_{2s}) = \sum_{i=1}^{M} (h_{1s}(i) log \frac{h_{1s}(i)}{m_i} + h_{2s}(i) log \frac{h_{2s}(i)}{m_i}) \tag{4.9}$$

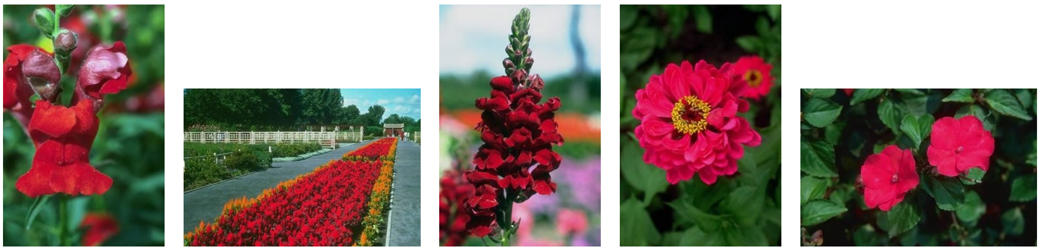
where $m_i = \frac{h_{1s}(i)+h_{2s}(i)}{2}$.

The results obtained by each implemented distance are showed in Figures 4.6, 4.7, 4.8, and 4.9 respectively. The results are very similar, independently of the similarity measure that had been chosen. All of them have retrieved images that are very similar to the query, so this descriptor seems to have an excellent behavior.



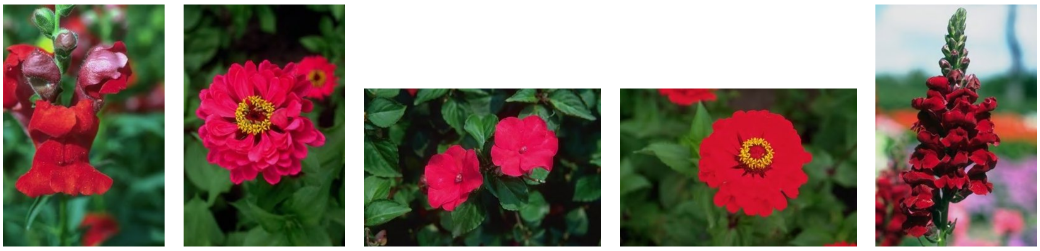
(a) query (b) retrieve #1 (c) retrieve #2 (d) retrieve #3 (e) retrieve #4

Figure 4.6: Results of query with L_1-norm similarity measure



(a) query (b) retrieve #1 (c) retrieve #2 (d) retrieve #3 (e) retrieve #4

Figure 4.7: Results of query with L_2-norm similarity measure



(a) query (b) retrieve #1 (c) retrieve #2 (d) retrieve #3 (e) retrieve #4

Figure 4.8: Results of query with histogram intersection similarity measure

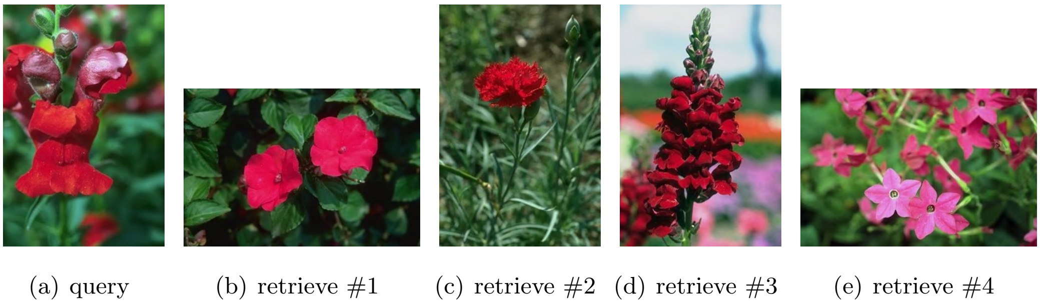

(a) query (b) retrieve #1 (c) retrieve #2 (d) retrieve #3 (e) retrieve #4

Figure 4.9: Results of query with Jeffrey divergence similarity measure

4.2.3 Color Layout Similarity Matching

For matching two Color Layout descriptors CLD_1 and CLD_2, each one defined as a set of luminance (DY) and chrominance (DCr and DCb) coefficients, the following distance measure can be used [MPS02]:

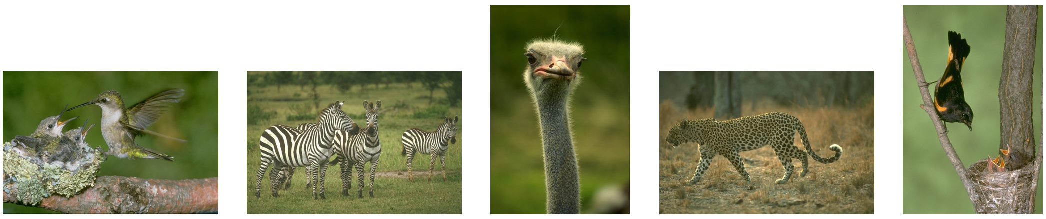
$$\begin{aligned} D(CLD_1, CLD_2) = & \sqrt{\sum_i w_{yi}(DY_{1i} - DY_{2i})^2} + \sqrt{\sum_i w_{bi}(DCb_{1i} - DCb_{2i})^2} \\ & + \sqrt{\sum_i w_{ri}(DCr_{1i} - DCr_{2i})^2} \end{aligned}$$

where DY_{ki}, DCb_{ki} and DCr_{ki} denote the i-th coefficients of Y, Cb, Cr color components of the Color Layout descriptor CLD_k and w_{yi}, w_{bi} and w_{ri} are the weighting values for the i-th coefficient, respectively. The distances should be weighted appropriately, with larger weights given to the lower frequency components, for visual purposes. Table 4.1 presents the recommended weighting values for each coefficient. They are designed to be implemented using only shift operations in order to accelerate the calculation speed [KY01]. If the number of coefficients is different between CLD1 and CLD2, the missing element values on the shorter descriptor should be regarded as 16, which means 0 value on AC coefficient fields before quantization process, or the redundant element values on the longer descriptor should be ignored.

Figure 4.10 shows an example in which the dissimilarity measure previously defined for Color Layout descriptor have been used. In this example, we can observe that the backgrounds of Figures 4.10(b), 4.10(c), and 4.10(e) are rather similar to the query's background.

	0	1	2	3	4	5
Y	2	2	2	1	1	1
Cb	2	1	1			
Cr	4	2	2			

Table 4.1: Recommended weighting values for Color Layout coefficients



(a) query (b) retrieve #1 (c) retrieve #2 (d) retrieve #3 (e) retrieve #4

Figure 4.10: Results of query with the standard similarity measure

4.3 Texture Descriptors Similarity Matching

In this section, the dissimilarity measure for the Texture Edge Histogram descriptor that is defined by the MPEG-7 standard is presented. Then, an illustrative example is given. Experimental results for this distance are given and analyzed in Chapter 6.

4.3.1 Texture Edge Histogram Similarity Matching

Although the 80 bins of the local-edge histogram are the standardized normative semantics for the Texture Edge Histogram descriptor, they alone may not be sufficient for image retrieval [MPS02]. Instead, some global-edge and semiglobal-edge directly computed from the local histogram are also used for an effective image matching.

For the global-edge histogram, the five edge categories for all subimages are accumulated. Similarly, for the semiglobal-edge histograms, there are 13 different subsets of subimages (see Figure 4.11) from which the corresponding edge histograms are generated. Each semiglobal histogram is obtained by accumulation of the bins of each type of edge corresponding to each subset, so a histogram of 65 bins is generated (13 subsets x 5 edge categories).

Combining the local, the semiglobal and the global histograms together, the following distance measure between two edge histogram descriptors A and B is given by [MPS02]:

$$D(A,B) = \sum_{i=0}^{79} |h_A(i) - h_B(i)| + 5 \text{ x } \sum_{i=0}^{4} |h^g_A(i) - h^g_B(i)| + \sum_{i=0}^{64} |h^S_A(i) - h^S_B(i)| \quad (4.10)$$

where $h_A(i)$ and $h_B(i)$ represent the normalized local edge histogram bin values of image A and image B, respectively, $h^g_A(i)$ and $h^g_B(i)$ represent the normalized bin values for the global-edge histograms, and $h^S_A(i)$ and $h^S_B(i)$ represent the histogram bin values for the semiglobal-edge histogram. In [MPS02], applying a weighting factor 5 for global-edge histogram in Equation 4.10 is recommended because the number of bins of this histogram is relatively smaller than the number of bins of local and semiglobal histograms. Some results obtained by this measure are showed in Figure 4.12. We can see than the retrieved images have the predominance of vertical edges in common with the query.

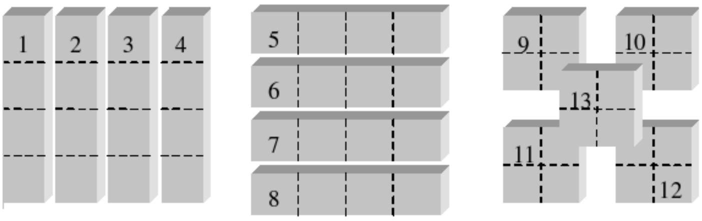
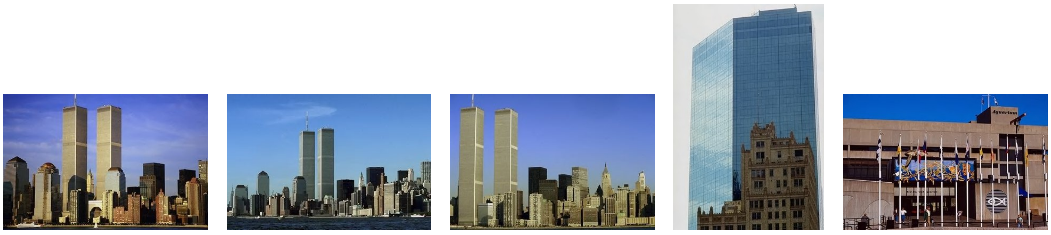


Figure 4.11: 13 subsets of subimages for semiglobal-edge histograms



(a) query (b) retrieve #1 (c) retrieve #2 (d) retrieve #3 (e) retrieve #4

Figure 4.12: Results of query with the standard similarity measure

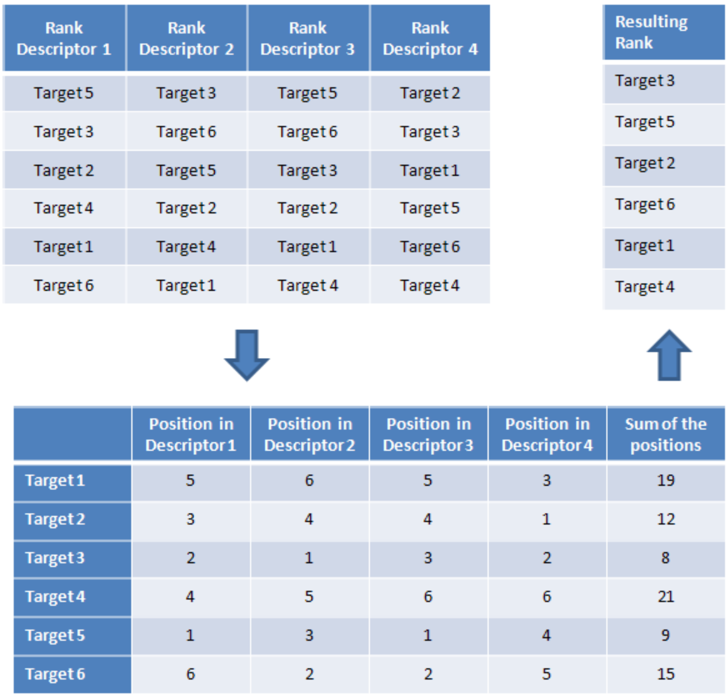
Chapter 5

Query by Example using a set of visual descriptors

The aim of this chapter is to take advantage of combining the several implemented descriptors instead of using only one. With this purpose, once the ranks are obtained for each desired descriptor we only take into account the positions in them for each target image. Therefore, the distance similarities previously computed are not considered any more.

As a consequence, the query by example using a set of visual descriptors consists of two steps. The first one consists in obtaining a rank for each descriptor. The dissimilarity measures presented in Chapter 4 are computed between the descriptors of the query image and the corresponding ones of each target image. Then, the target images are ordered by increasing order of dissimilarity measure for each descriptor. So, the first element in a rank descriptor corresponds to the target image that is the most similar one to query image for this descriptor.

The second step consists in obtaining a new rank resulting from the previous ones, which have been computed for each descriptor. In order not to have to normalize each distance, which would suppose to study the probability density function for each dissimilarity measure, we have decided to take into account only the position of each target image in each descriptor rank. Therefore, having a set of N_d descriptors and the ranking of each target image for each of them, the distance gathering all the



Rank Descriptor 1	Rank Descriptor 2	Rank Descriptor 3	Rank Descriptor 4
Target 5	Target 3	Target 5	Target 2
Target 3	Target 6	Target 6	Target 3
Target 2	Target 5	Target 3	Target 1
Target 4	Target 2	Target 2	Target 5
Target 1	Target 4	Target 1	Target 6
Target 6	Target 1	Target 4	Target 4

Resulting Rank
Target 3
Target 5
Target 2
Target 6
Target 1
Target 4

	Position in Descriptor 1	Position in Descriptor 2	Position in Descriptor 3	Position in Descriptor 4	Sum of the positions
Target 1	5	6	5	3	19
Target 2	3	4	4	1	12
Target 3	2	1	3	2	8
Target 4	4	5	6	6	21
Target 5	1	3	1	4	9
Target 6	6	2	2	5	15

Figure 5.1: Process to obtain the resulting rank

descriptors is defined as:

$$D(Q, T_i) = \sum_{k=1}^{N_d} w_k \, p_{ik} \tag{5.1}$$

where p_{ik} is the position of the target image T_i in the rank corresponding to the k-th descriptor and w_k is its associated weight. These weights allow the user to give more importance to some descriptors, making the application more flexible. Eventually, the target images are ordered by increasing order of Equation 5.1 and the rank resulting from the set of visual descriptors is obtained. The whole extraction process is showed by an illustrative example in Figure 5.1.

The next section give some details about the tool that has been implemented in order to retrieve images by using either an isolated descriptor or a set of descriptors, as well the graphical user interface used to show the results.

5.1 Implemented tool

Regarding the implementation, a tool that is called RANKER has been designed. This application requires three arguments:

- Query image file: It must be a file of XML format. This file can be given in two different ways:
  - Descriptor file: It is a file whose name is ending by “-vd.xml”. This file contains all the computed descriptors for the query image.
  - Pointer file: It is a file whose name is ending by “-gos.xml”. This file contains the ubication of the descriptor file in which there are all the computed descriptors for the query image. Furthermore, the region identifier is specified, which it will be the corresponding one to the root node because the whole image is considered.
- Configuration file: It must be an XML format file. The descriptors that have to be considered are specified in it with its corresponding weight in order to generate the resulting rank. It contains the ubication of the database searching space as well. The file represented by this ubication is a TXT file which contains the ubication of the descriptor file for each target image. There are two other attributes. One of them specifies which is the number of results to be given. The other one refers to the fusion method that has been used.
- Results directory: It indicates where the results will be stored.

Then, the application extracts the descriptor files for each target image specified in the TXT file and computes the dissimilarity measures between them and the query image for each descriptor specified in the configuration file. It results in a rank for each descriptor which are combined in order to obtain only one rank for the set of descriptors.

Finally, the top of the rank representing the target images most similar to the query image are stored in a XML file, which is created in the ubication given by the results directory. The number of results which have to be stored are specified in the configuration file. The query image filename, the visual descriptors and its weights, and the database that contains the target images’ filenames are also specified in it.

Since the objective of the present work is the image retrieval, all implemented descriptors will not be used. In particular, the descriptors that are based on shape feature have only sense for regions. Therefore, the following descriptors whose dissimilarity measures have been analyzed can be used in the implemented application:

- Color Structure Descriptor
- Dominant Color Descriptor
- Color Layout Descriptor
- Texture Edge Histogram Descriptor

As commented in Chapter 2, this application has been integrated in a graphical interface, called GOS (Graphic Object Searcher), in order to make easier the interaction with the user. This program allows the user to set some parameters of configuration such as the query image, the number of results, the search space, the desired descriptors and their weights in an easier way. A capture of this program is shown in Figure 5.2.

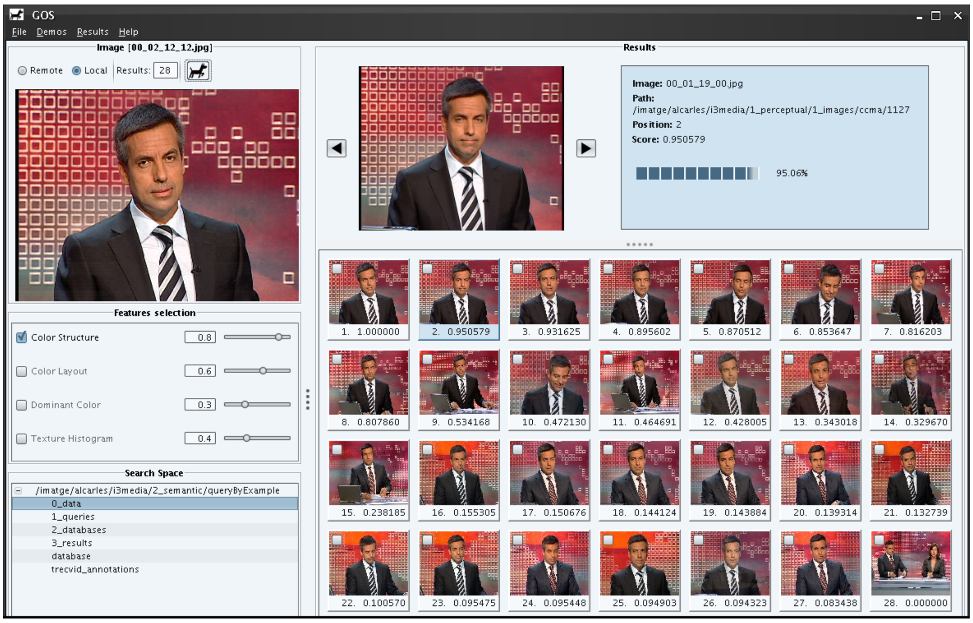


Image [00_02_12_12.jpg]

○ Remote ● Local Results: 28

Features selection

☑ Color Structure 0.8
□ Color Layout 0.6
□ Dominant Color 0.3
□ Texture Histogram 0.4

Search Space

/imatge/alcarles/i3media/2_semantic/queryByExample
0_data
1_queries
2_databases
3_results
database
trecvid_annotations

Results

Image: 00_01_19_00.jpg
Path: /imatge/alcarles/i3media/1_perceptual/1_images/ccma/1127
Position: 2
Score: 0.950579

95.06%

1. 1.000000 | 2. 0.950579 | 3. 0.931625 | 4. 0.895602 | 5. 0.870512 | 6. 0.853647 | 7. 0.816203
8. 0.807860 | 9. 0.534168 | 10. 0.472130 | 11. 0.464691 | 12. 0.428005 | 13. 0.343018 | 14. 0.329670
15. 0.238185 | 16. 0.155305 | 17. 0.150676 | 18. 0.144124 | 19. 0.143884 | 20. 0.139314 | 21. 0.132739
22. 0.100570 | 23. 0.095475 | 24. 0.095448 | 25. 0.094903 | 26. 0.094323 | 27. 0.083438 | 28. 0.000000

Figure 5.2: A capture of GOS program

Chapter 6

Experimental Results

For evaluation of retrieval performance, MPEG group have defined an evaluation metric called Averaged Normalized Modified Retrieval Rate (ANMRR) in order to measure the performance of retrieval. It was developed on the basis of the specification of a data set, a query set and the corresponding ground-truth data, which is a set of visually similar images for a given query image.

In defining ANMRR, the following factors were considered:

- The measure should be normalized to account for the variation in size of the ground truth among the different queries.
- The measure should favor algorithms that retrieve the ground-truth items as the top matches.
- The measure should assign a penalty for each of the missed ground-truth items. If a ground-truth item is not retrieved within a certain number of top matches, then it is considered as missed. The penalty should be selected such that beyond a certain limit on the rank, it should not matter whether a ground-truth item is found or not, for example, at the 200th or at the 2000th rank.
- The measure should consider the order in which the ground-truth items are retrieved, so it should favor algorithms that retrieve ground-truth items in highest ranks.

The following solution was adopted. Consider a query q with a ground-truth size of $NG(q)$; the rank $\mathbf{Rank}(k)$ of the kth ground-truth image is defined as the position at which this ground-truth item is retrieved (a rank value of one corresponds to the top match). Furthermore, a number $K(q) \geq NG(q)$ is defined, which specifies the relevant ranks, that is, retrieval with rank larger than $K(q)$ should be considered as a *miss*. In order to penalize the misses images, it was decided to define the $\mathbf{Rank}(k)$ as:

$$\mathbf{Rank}(k) = \begin{cases} \mathbf{Rank}(k) & \text{if} \quad \mathbf{Rank}(k) \leq K(q) \\ 1.25 \cdot K(q) & \text{if} \quad \mathbf{Rank}(k) > K(q) \end{cases} \tag{6.1}$$

where a suitable $K(q)$ is determined in [MPS02] by:

$$K(q) = \min\{4 \cdot NG(q), 2 \cdot \max[NG(q), \forall q]\}$$

From Equation 6.1 the Average Rank (AVR) for query q is computed by:

$$\mathbf{AVR}(q) = \frac{1}{NG(q)} \sum_{k=1}^{NG(q)} \mathbf{Rank}(k)$$

However, with ground-truth sets of different size, the AVR counted from ground-truth sets with small and large $NG(q)$ values would largely differ. In order to eliminate influences of different $NG(q)$, the *Modified Retrieval Rank* is defined as:

$$\mathbf{MRR}(q) = \mathbf{AVR}(q) - 0.5 \cdot [1 + NG(q)]$$

The MRR is always larger than or equal to 0, but with upper bound still dependent on $NG(q)$. The worst scenario is that no ground-truth image has been retrieved. In this case, the value of AVR would be $1.25K(q)$, so the maximum value for MRR, which is given in this scenario, is $1.25K(q) - 0.5[1 + NG(q)]$. This finally leads to the *Normalized Modified Retrieval Rank*:

$$\mathbf{NMRR}(q) = \frac{\mathbf{MRR}(q)}{1.25 \cdot K(q) - 0.5 \cdot [1 + NG(q)]} \tag{6.2}$$

Now, the NMRR(q) can take values between 0 (indicating whole ground-truth retrieved as the top matches) and 1 (indicating nothing found), irrespective of $NG(q)$. From Equation 6.2, it is straightforward to define the *Average Normalized Modified Retrieval Rate* (**ANMRR**), giving just one number indicating the retrieval quality over all queries.

$$\mathbf{ANMRR}(q) = \frac{1}{NQ}\sum_{q=1}^{NQ}\mathbf{NMRR}(q)$$

where NQ is the number of queries.

An experiment performed in [NRM$^+$00] evidences that the ANMRR measure approximately coincides linearly with the results of subjective evaluation about retrieval accuracy of search engines. From this experiment, the categorical relationship between ANMRR and the subjective ratings has been estimated, which is given in Table 6.1.

Subjective Ratings	**ANMRR**
Very good - Good	0 - 0.24
Good - Fair	0.24 - 0.41
Fair - Poor	0.41 - 0.58
Poor - Very poor	0.58 - 1

Table 6.1: Categorical relationship between ANMRR and the subjective ratings

The image dataset used is the Common Color Dataset (CCD) with Common Color Query (CCQ) defined by MPEG group in [ZO99], which contains 5466 images and 50 queries with ground truth set. This dataset is used in order to obtain the ANMRR for each visual descriptor. CCD is the dataset used in MPEG-7 conformance test for color descriptors, so we will be able to compare the results from [MPS02].

The CCD consists of a variety of still images, with 7 subsets (s1, s3, add1, add2, add3, add4, add5) from different sources, including images from stock photo galleries (s1, s3, add3), images extracted from TV programs and other video sequences (add1, add2, add4, add5). In Table 6.2 there is the list of ground truth sets and their parameters. The details of ground-truth sets are available in Appendix C.

Query Name	NG(q)	K(q)
1. Flower garden	4	16
2. Rock and sky	6	24
3. NEWS anchor	10	40
4. Walking people	17	64
5. Baldheaded man walking and talking with persons	24	64
6. Sports reporters in the rain	9	36
7. Congress	8	32
8. Baldheaded man representing	32	64
9. Castle	7	28
10. Black clothes lady on the blue mat	6	24
11. Singer with studio lights	5	20
12. Strange hair	5	20
13. Leather jacket people	10	40
14. Man with placard	5	20
15. People on the red	5	20
16. Snake	12	48
17. Fish	11	44
18. Tapirs	12	48
19. Butterfly	11	44
20. Small monkey with banana	12	48
21. Landscape Image 1	4	16
22. Landscape Image 2	4	16
23. Landscape Image 3	3	12
24. Indoor Image	12	48
25. Anchorperson	17	64
26. Quiz Scene	4	16
27. Speaker	6	24
28. Man and horse	6	24
29. Space earth	4	16
30. Fountain	7	28
31. Graphics before NEWS	9	36
32. Ron Reagan	9	36
33. Basketball GAME overlay	4	16
34. Glass roof	4	16
35. Snow clad mountain	4	16
36. Outdoor/boats	6	24
37. By the water	4	16
38. Couple	5	20
39. Shop	6	24
40. Flower (indoor)	4	16
41. Playing on the street	5	20
42. Road with trees/grass	4	16
43. Children/rock/grass	4	16
44. Asian building	5	20
45. Containers	4	16
46. Sunset over lake	8	32
47. Big pipes	6	24
48. Man with sunglasses in white shirt	3	12
49. Wooden shack	6	24
50. Ruins	8	32

Table 6.2: MPEG-7 CCD

Now, the results obtained by each visual descriptor in the CCD will be analyzed. The details of the results for each query belonging to the CCQ are showed in Table 6.6. Each value of this table represents the NMRR obtained by the query image indicated by the row and the dissimilarity measure for the descriptor indicated by the column.

Relating to the *Dominant Color Descriptor* (DCD), the results for the following dissimilarity measures have been obtained:

- MPEG-7 (DCD MPEG7). This is the distance proposed by the MPEG-7 group that does not take account into the spatial coherency and the color variance parameters. It has been used with a penalization for the case that no dominant color values have been matched between the query and target images.
- Percentage (DCD PERC). This is a variation of the previous distance in which a minimum percentage of the dominant color value is required to be matched. In the experimental results, the percentage has been set to 50.
- Merged Palette Histogram Similarity Matching (DCD MPHSM). This is the distance proposed in [PW04] without introducing any modifications.
- MPHSM-Hungarian (DCD MPHSM-H). This is the previous distance in which the Hungarian algorithm has been used when generating the color merged palette.

Table 6.3 shows a comparison of ANMRR results for these dissimilarity measures.

DCD MPEG7	DCD PERC	DCD MPHSM	DCD MPHSM-H
0.3410	0.3368	0.5622	0.4173

Table 6.3: ANMRR results for Dominant Color Descriptor

According to these results, the distance proposed by MPEG-7 and its variation including a minimum percentage give the best matches (0.3410 and 0.3368, respectively). The difference between them is insignificant because they are only 0.0042 distance apart. However, for some queries, the results obtained by the DCD PERC are slightly better than the DCD MPEG7 ones. Figure 6.1 shows an example of the retrieved images using the distance proposed by MPEG-7 while the improved results by guaranteeing a minimum percentage are presented in Figure 6.2. In this example, the query image #32 has been used. The images that are marked by green rectangle boxes belong to the corresponding ground truth. Thus, in Figure 6.1 we can see that 7 of 9 ground-truth images have been retrieved, whereas 8 of 9 ground-truth images have been found in Figure 6.2. Therefore, the NMRR value obtained by this latter dissimilarity measure (0.1889) is slightly better than the given by the first one (0.2222).

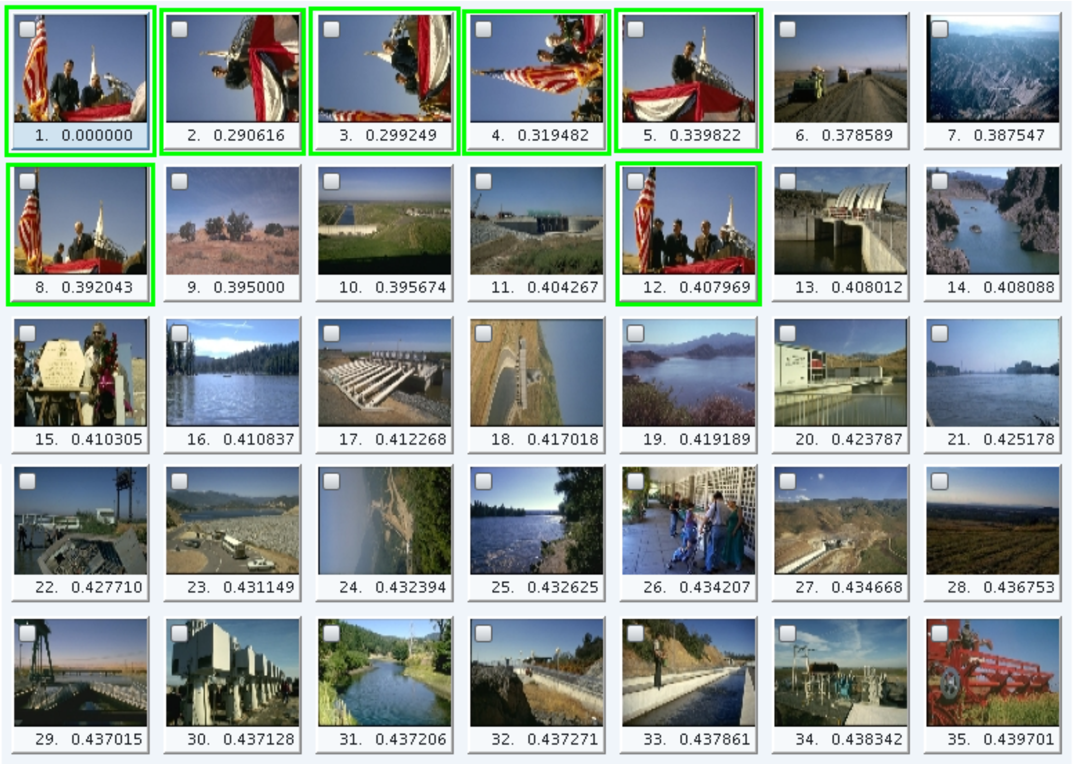


Figure 6.1: Retrieved images for query #32 using DCD MPEG7

From the results obtained in the example showed in Figure 6.3, which corresponds to the query #13, we can observe that, in some cases, the ground truth given by MPEG-7 is too strict because most of the retrieved images appearing on the top of the rank could be consider similar to the query image and be classified in the same category (singer with studio lights). From my point of view, the images that are ranked in 2, 5, 6, 7, 9, 11, and 12 positions could belong to the ground-truth set.

Although the MPHSM is presented in [PW04] as an improved dissimilarity measure for the Dominant Color Descriptor, the results obtained using this dataset are the worst ones (0.5622). However, the ANMRR decrease when the hungarian algorithm

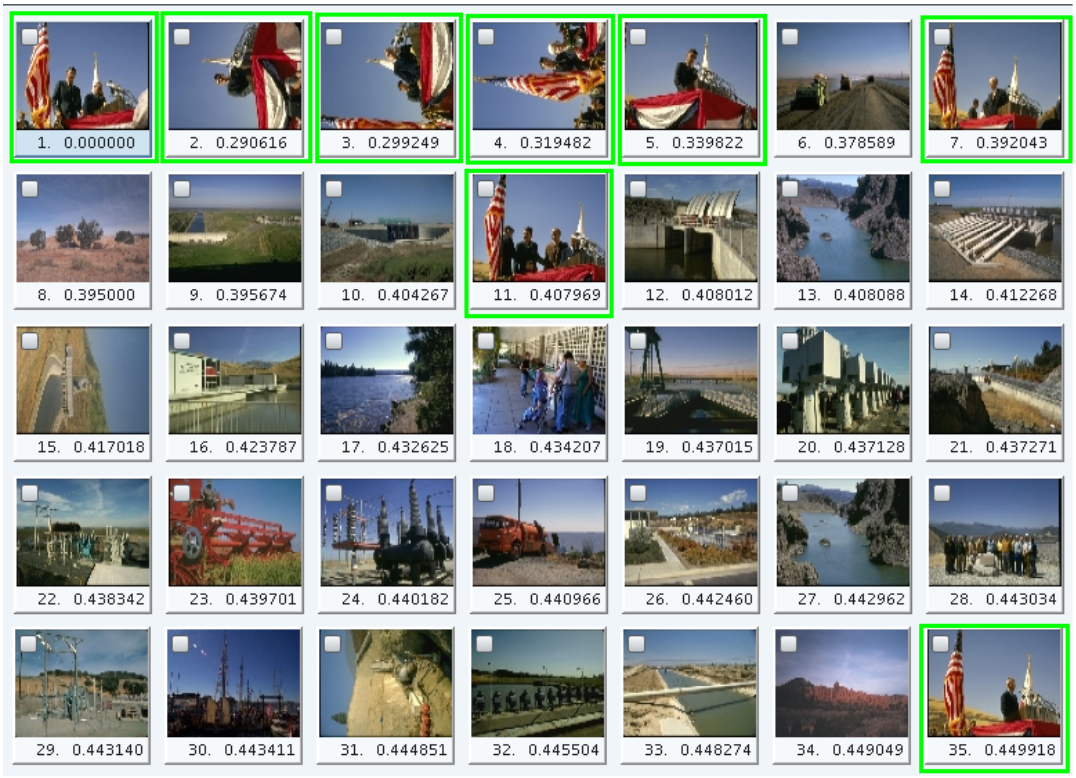


Figure 6.2: Retrieved images for query #32 using DCD PERC

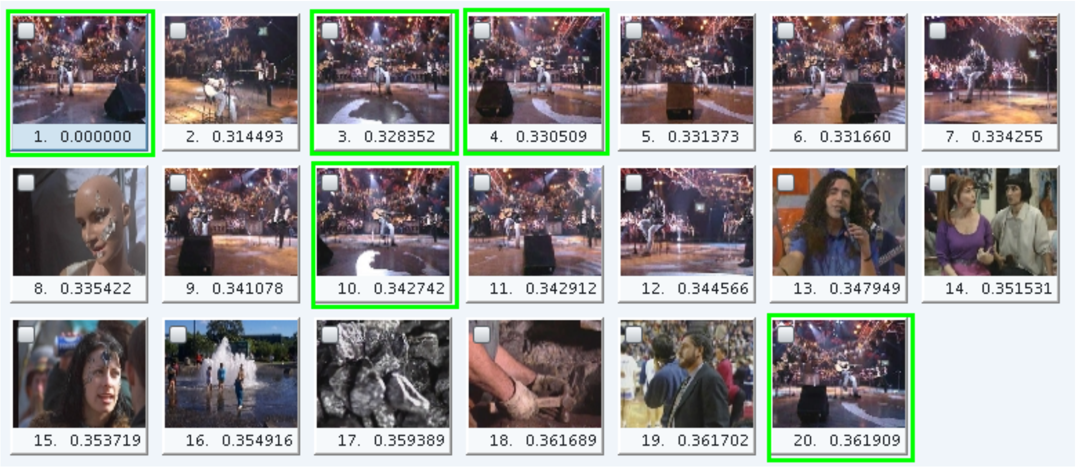


Figure 6.3: Retrieved images for query #13 using DCD MPEG7 with a strict ground truth

is used (0.4173). Analyzing the NMRR obtained for each query, only the 10% of them are matched better with the retrieved images if the hungarian method is not used. Moreover, for 20% of the queries, the results given by this measure are the best between the all implemented distances. An example of the images retrieved by these two dissimilarity measures are showed in Figures 6.4 and 6.5. For both examples, the image query #3 has been used. When the MPHSM is applied, 7 of 10 ground-truth images are retrieved, whereas 9 ground-truth images are found when the hungarian algorithm is introduced. In spite of this improvement, the distance proposed by MPEG-7 including the minimum percentage option gives the best results, so this is the best choice for the Dominant Color Descriptor. Moreover, the value obtained for ANMRR is rather close to the resulting one from MPEG-7 experiments (0.31) [MPS02].

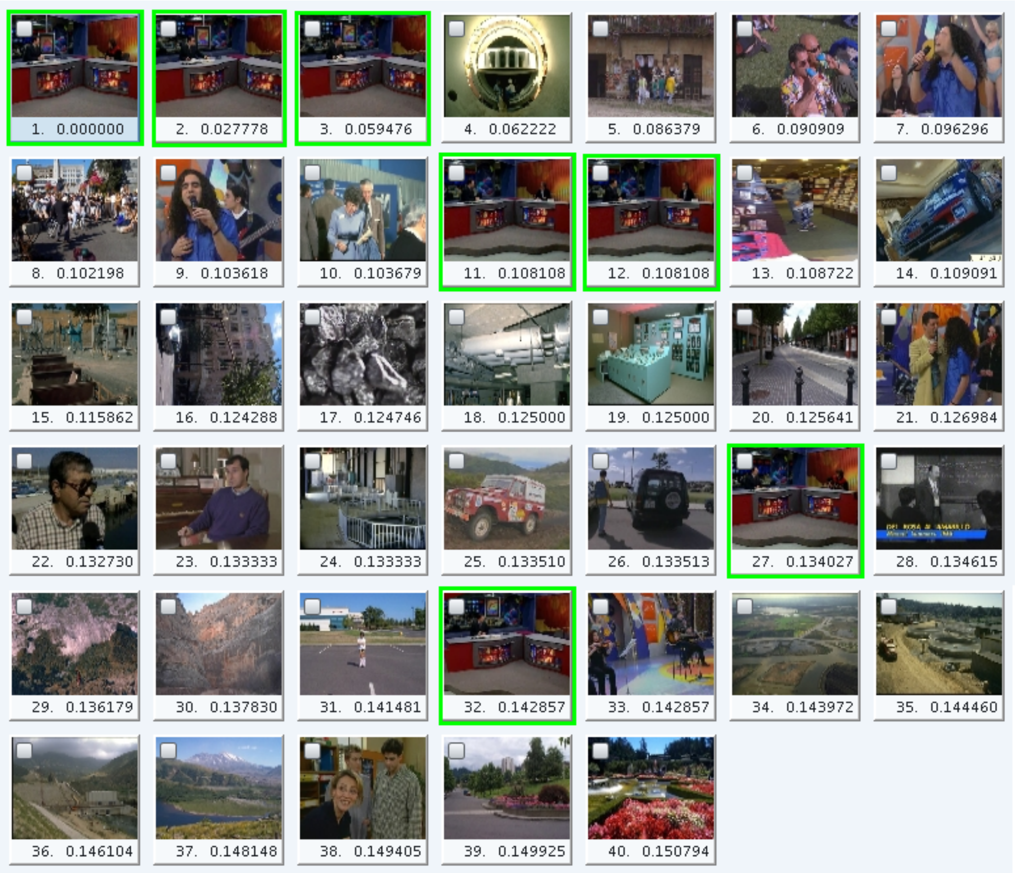


Figure 6.4: Retrieved images for query #3 using DCD MPHSM

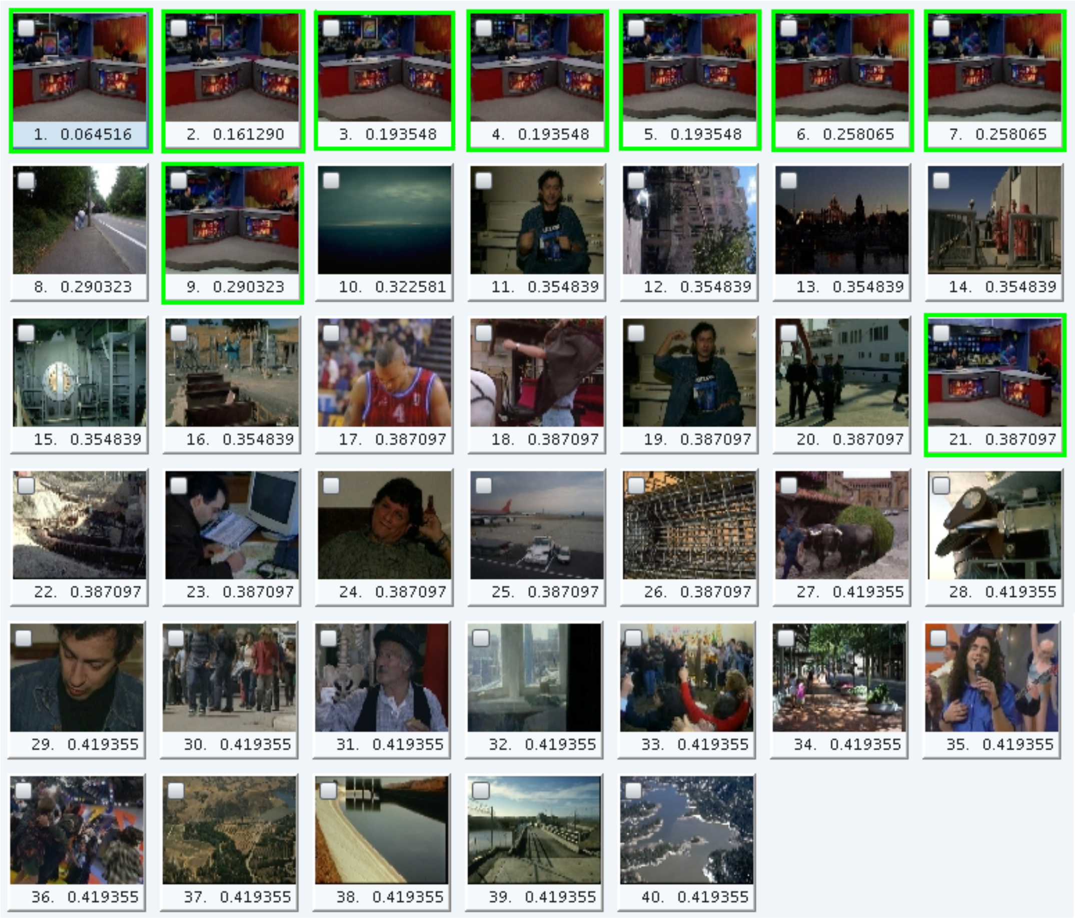


Figure 6.5: Retrieved images for query #3 using DCD MPHSM-H

Regarding the *Color Structure Descriptor*, the results for the following dissimilarity measures have been obtained:

- L_1-norm (CSD L1). This is the distance proposed by the MPEG-7 group.
- L_2-norm (CSD L2). This is associated with the Euclidean distance.
- Jeffrey divergence (CSD JF). Distance that takes advantage of the information theory and probability theory properties.
- Intersection Histogram distance (CSD IH). It computes the intersection area between two histograms.

According to the results showed in Table 6.4, the Jeffrey divergence gives the best result (0.0234). This dissimilarity measure improves the result given by MPEG-7

CSD L1	CSD L2	CSD JF	CSD IH
0.0426	0.0696	0.0234	0.1817

Table 6.4: ANMRR results for Color Structure descriptor

experiments in [MPS02], in which a value of 0.06799 was obtained. The NMRR obtained for each query in the Common Color Query is only beaten by the L_1-norm in the 12% of the cases. For the rest of cases, the Jeffrey divergence gives the best performance. In Figure 6.6, an example of the retrieved images for this dissimilarity measure is showed. In this example, the image query #28 has been used and all ground-truth images have been retrieved. The resulting NMRR for this query is 0.0189, a value rather close to the ANMRR obtained using this dissimilarity measure for the whole set of queries (0.0234).

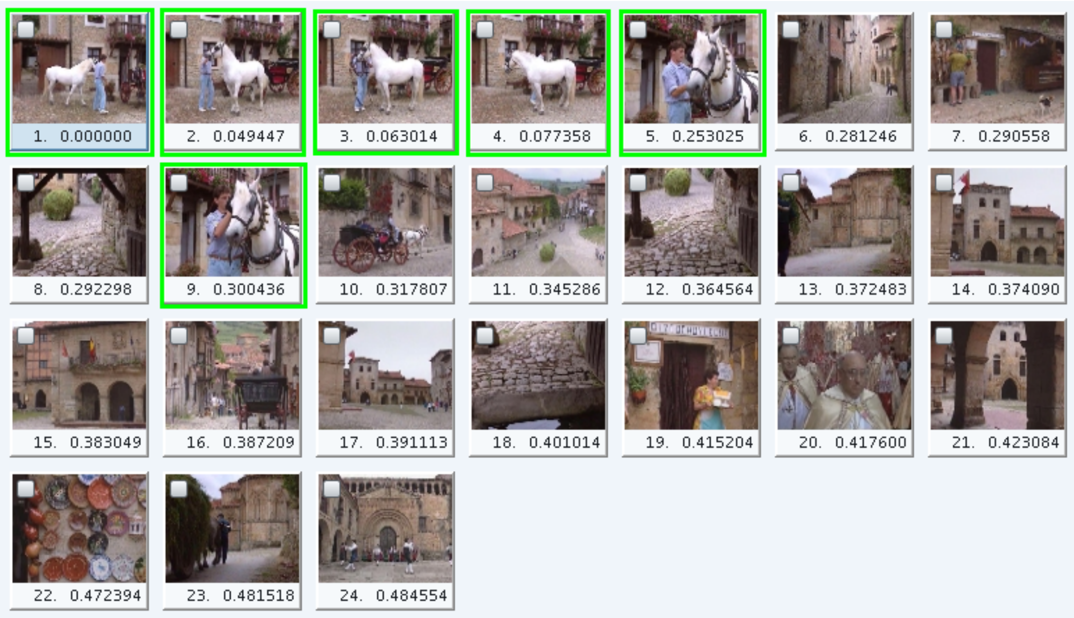


Figure 6.6: Retrieved images for query #28 using CSD JF

Although the results obtained by L_1-norm are slightly worse (0.0426), the ANMRR resulting from the CCQ in our experiments is still better than the one given by MPEG-7 group (0.06799). In Figure 6.7, the retrieved images by the same query showed in the previous figure are presented. Like the previous example, the whole ground truth have been found but some images are worse positioned in the rank.

Thus, the NMRR value obtained by this query is 0.1195, which is not as good as the given by the Jeffrey divergence (0.0189).

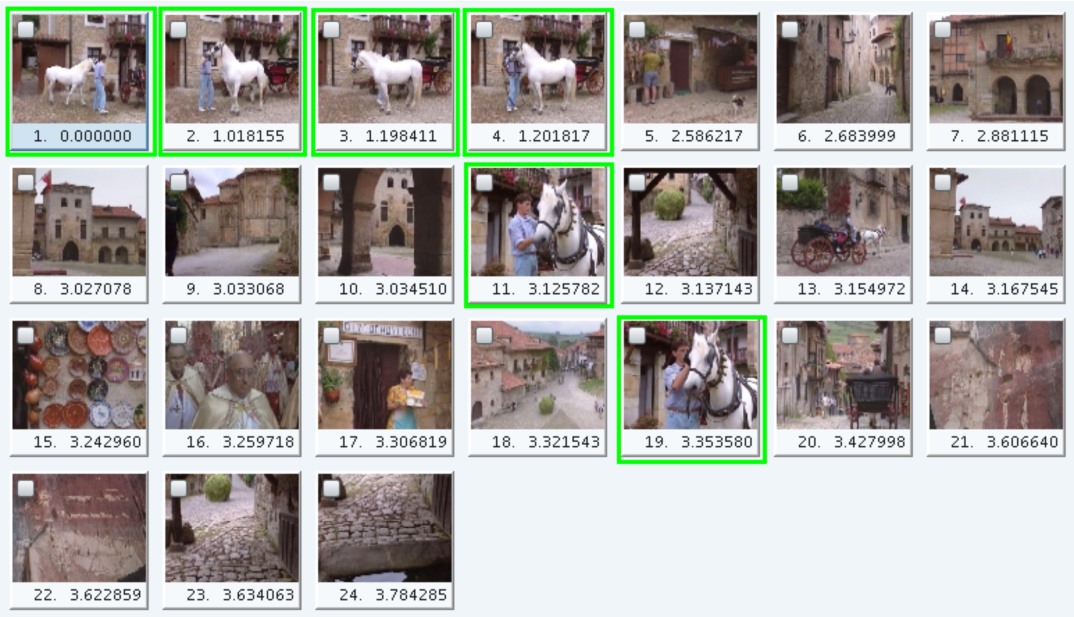


Figure 6.7: Retrieved images for query #28 using CSD L1

Using L_2-norm, a similar ANMRR to the MPEG-7 result for Color Structure descriptor has been obtained (0.0696). Therefore, this dissimilarity measure also has a very good behaviour. On the other hand, the results given by the histogram intersection distance are clearly the worst ones (0.1817). Figures 6.8 and 6.9 show some examples of retrieval images obtained using these two dissimilarity measures. The results obtained in the second figure are worse because one ground-truth image has not been found, whereas in the first one the whole ground-truth set has been retrieved. From the analysis of color structure results, there is no doubt that the Jeffrey divergence has to be chosen for the image retrieval application.

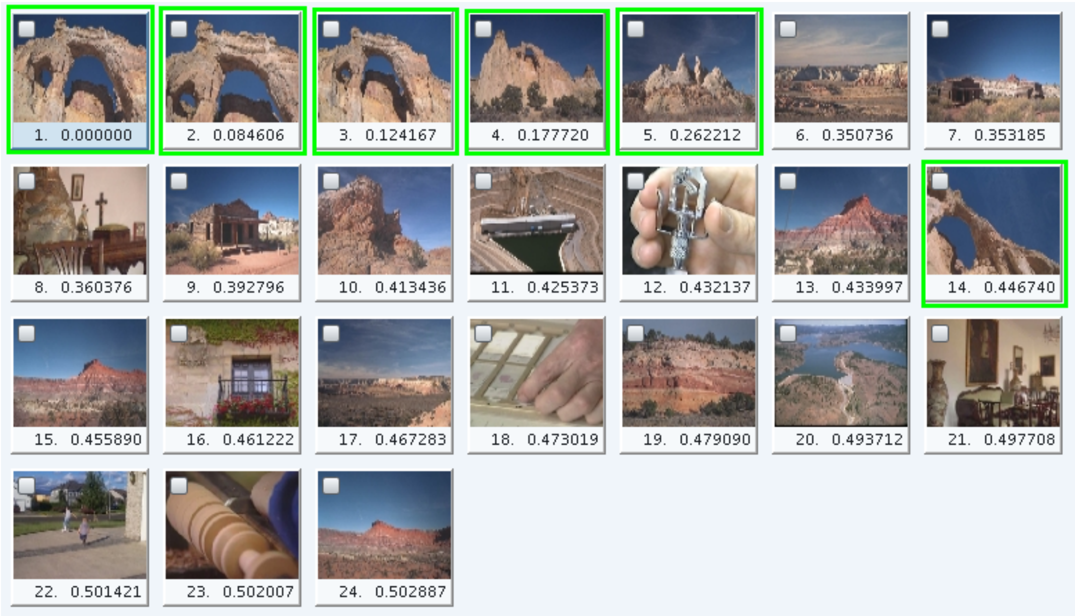


Figure 6.8: Retrieved images for query #2 using CSD L2

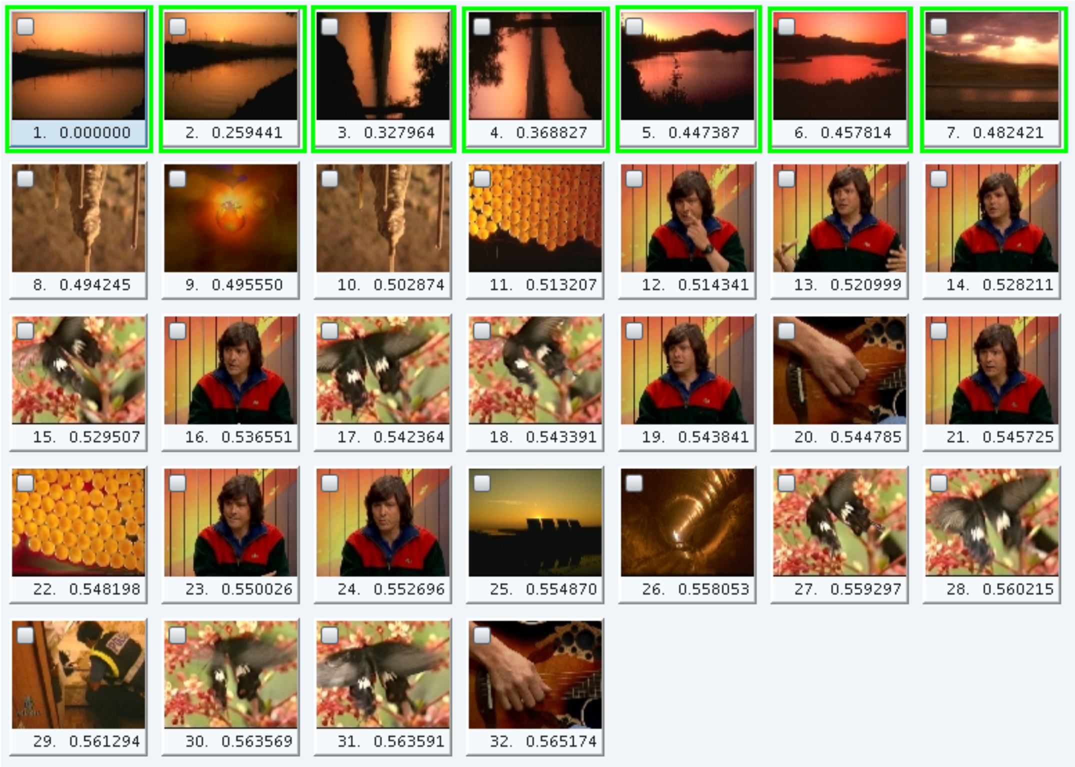


Figure 6.9: Retrieved images for query #46 using CSD IH

Relating to the *Color Layout Descriptor*, the ANMRR obtained from the CCD is 0.2350, which is very close to the value extracted from MPEG-7 experiments (0.22). According to Table 6.1, an ANMRR rate about 0.24 could be considered as a good result. Furthermore, this descriptor is very compact although the results are not so good as the ones obtained by the Color Structure descriptor. Therefore, it is good for small storage and fast retrieval. In Figure 6.10, an example of the retrieved images for the query #36 is presented, in which all ground-truth images have been found as top matches. For this query in particular, the results given by the Color Layout descriptor are the best ones.

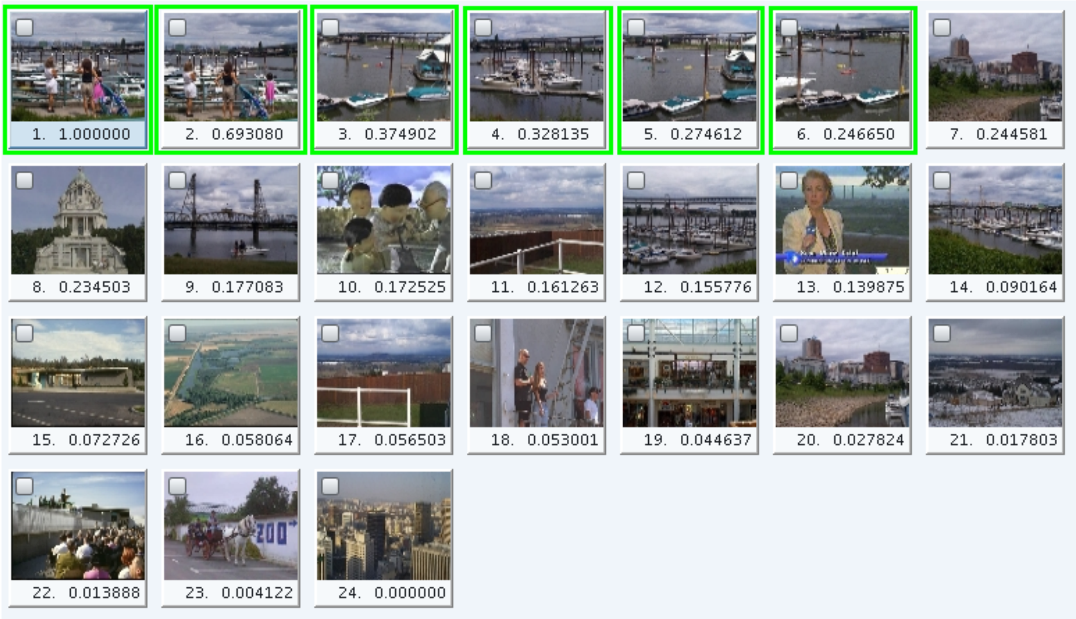


Figure 6.10: Retrieved images for query #36 using CLD

Regarding the *Texture Edge Histogram Descriptor*, in the MPEG-7 Core Experiments [CER99], a dataset containing 11639 images, which is more appropiated for this descriptor, was used. The ANMRR value obtained by MPEG-7 in this database is 0.2969. Since we could not get this dataset, the CCD and the CCQ were used in order to obtain the results for this descriptor. In spite of using a different database, the ANMRR obtained from the Texture Edge Histogram descriptor, whose value is 0.3281, is rather close to the one given by MPEG-7 (0.2969) in [PJW00]. The retrieved images for query #42 using this descriptor are showed in Figure 6.11. For this query, the whole ground-truth set has been found and the obtained NMRR is 0.1429.

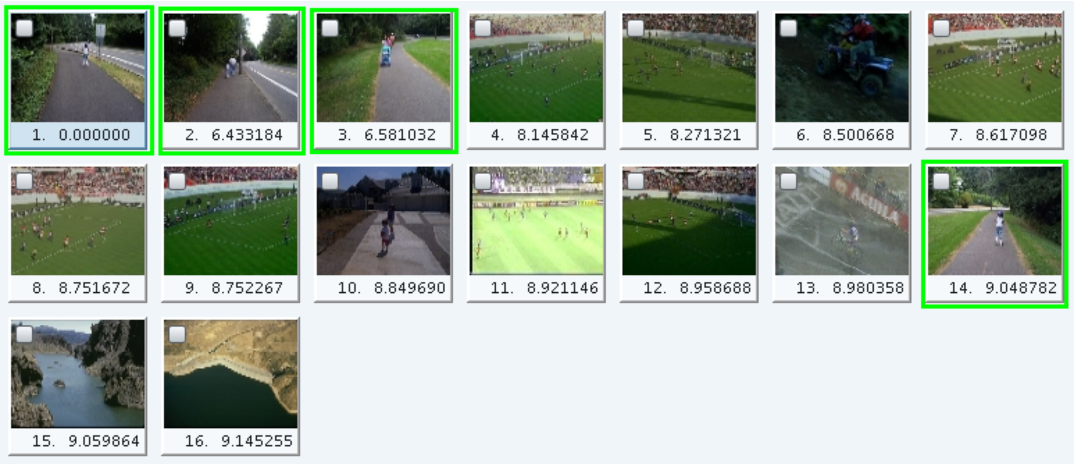


Figure 6.11: Retrieved images for query #42 using EHD

Once the results have been analyzed independently for each visual descriptor, we will try to improve them by fusioning some descriptors. Because of the very good results given by the Color Structure descriptor in the used database, we cannot expect that the results combining all descriptors will be improved. This fact is explained because if a method is far better than the other ones used, then the latter ones are detrimental to the results obtained by the best one. For example, if a ground-truth image was ranked in the first position for one descriptor and in the 50th position for the other three descriptors it would obtain a worse score than a non-ground-truth image that was ranked in the 30th position for all of them. However, it would be expected that the results obtained by fusioning them would be better than the given by the worst ones. Since this is not the expected behavior, we have decided not to use the best descriptor in order to show that combining the results obtained by three similar descriptors, which have a similar ANMRR, we can obtain some better results.

Thus, we have fusioned the Dominant Color, the Color Layout, and the Texture Edge Histogram descriptors. The weights are the same for each of them, so we have not given more importance to one descriptor than the others. The resulting ANMRR obtained by the fusion (DCD+CLD+EHD) is compared to the ones given by each of them in Table 6.5. The comparison between the NMRR obtained for each query image of the CCD are detailed in Table 6.7.

From the analysis of these results, some statistics can be extracted. In a 60% of the cases, the NMRR value obtained by fusioning the three visual descriptors is better

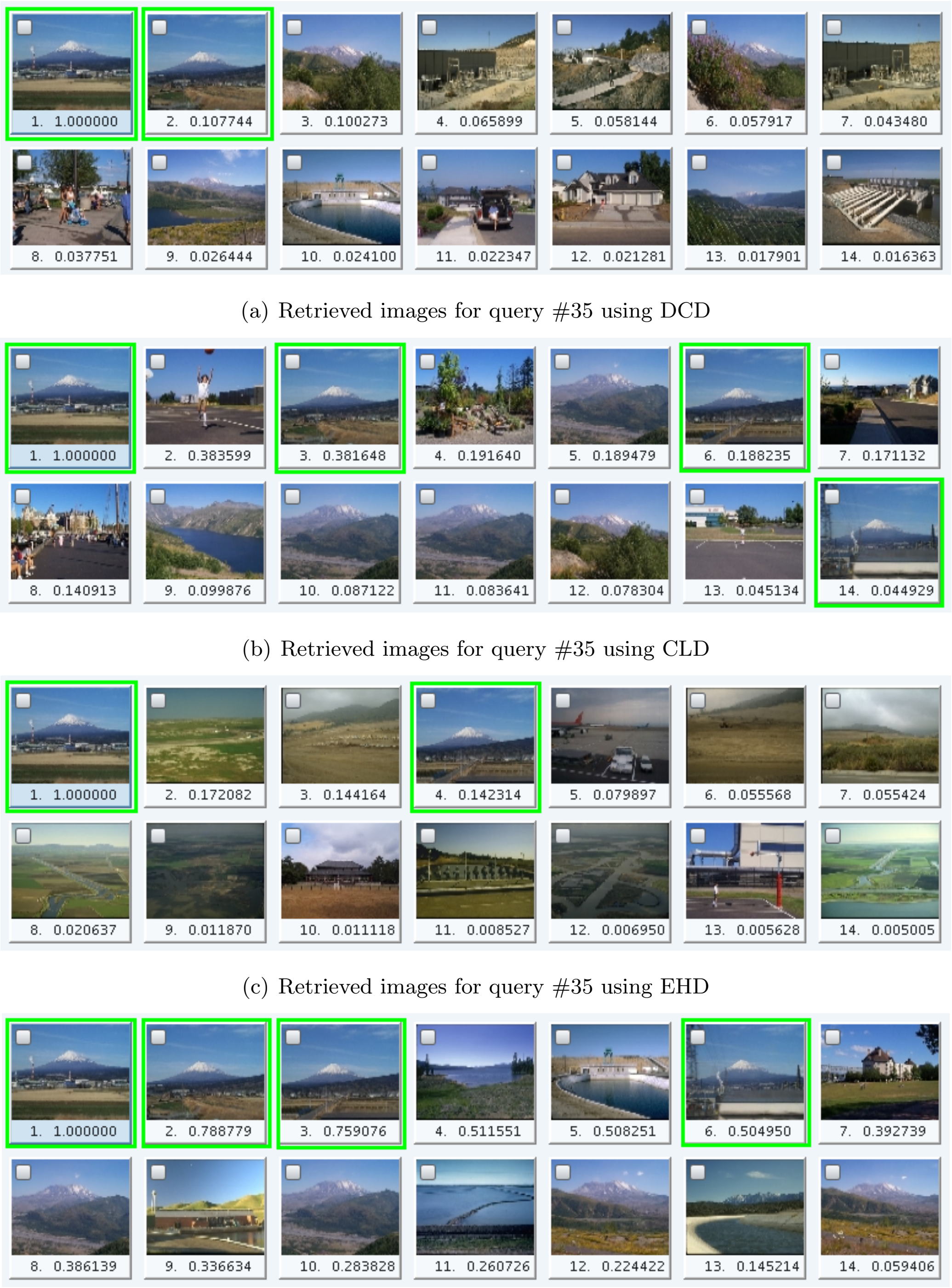
DCD	CLD	EHD	DCD+CLD+EHD
0.3368	0.2350	0.3281	0.1872

Table 6.5: ANMRR results obtained by fusion

than the NMRR values given by each descriptor independently. Therefore, the fusion of descriptors can lead to a better behavior than the obtained by each of them. Furthermore, only in a 12% of the queries belonging to CCQ, the resulting NMRR coincides with the worst one between the three descriptors.

In Figure 6.12, an example of improving the behaviour of some descriptors fusioning them is showed. Figures 6.12(a), 6.12(b), and 6.12(c) depict the retrieved images for query #35 using the Dominant Color, the Color Layout, and the Texture Edge Histogram descriptors, respectively. All the ground-truth images have been retrieved using the Color Layout Descriptor, whereas only two of the four ground-truth images have been found when Dominant Color and Texture Edge Histogram are used. Figure 6.12(d) shows the retrieved ones when the fusion of these descriptors has been used. Although in 6.12(b) all the expected images have been retrieved, their ranks (1, 3, 6, 14) are not as good as the ones obtained by the fusion of these descriptors (1, 2, 3, 6). Thus, the NMRR reached in this latter case is 0.0286, whereas the values obtained for each descriptors are 0.4714, 0.2000, and 0.5000, respectively.

If the Color Structure descriptor is also used, it has been experimented that the improvement is not significant. This shows that the arguments previously discussed are true. The NMRR has decreased to 0.1615 but this value is not nearly as good as the NMRR obtained using only this descriptor.



(a) Retrieved images for query #35 using DCD

(b) Retrieved images for query #35 using CLD

(c) Retrieved images for query #35 using EHD

(d) Retrieved images for query #35 by fusion of DCD, CLD and EHD

Figure 6.12: Comparison between retrieved images using only a descriptor and using a fusion of them for query #35

Query	DCD	DCD	DCD	DCD	CSD	CSD	CSD	CSD	CLD	EHD
	MPEG7	PERC	MPHSM	MPHSM-H	L1	L2	JF	IH		
1	0.7286	0.7286	0.7286	0.7286	0.0000	0.0000	0.0000	0.6571	0.5429	0.5286
2	0.4528	0.4528	0.4214	0.3459	0.0000	0.0503	0.0000	0.0000	0.5472	0.4717
3	0.0517	0.0517	0.4112	0.1191	0.0000	0.0000	0.0000	0.0000	0.0472	0.0045
4	0.0820	0.1218	0.5642	0.4383	0.0000	0.0000	0.0000	0.0000	0.0729	0.3471
5	0.1944	0.1920	0.6500	0.4210	0.0000	0.0000	0.0000	0.0006	0.0019	0.0160
6	0.1194	0.1194	0.6139	0.2806	0.0028	0.0056	0.0028	0.1194	0.1500	0.2861
7	0.0599	0.0599	0.6338	0.5317	0.0528	0.0669	0.0141	0.4120	0.2500	0.4401
8	0.1088	0.1073	0.5207	0.2426	0.0000	0.0000	0.0000	0.0876	0.1491	0.1693
9	0.0000	0.0000	0.6498	0.0046	0.0000	0.0046	0.0000	0.3272	0.0000	0.1290
10	0.0377	0.0377	0.5220	0.0566	0.0000	0.0063	0.0000	0.0000	0.0063	0.1698
11	0.2091	0.0727	0.7818	0.3727	0.0636	0.0455	0.0273	0.2364	0.0909	0.0273
12	0.1091	0.1091	0.3818	0.3727	0.0000	0.0091	0.0000	0.0091	0.1000	0.3727
13	0.6584	0.6584	0.8247	0.6899	0.0427	0.1011	0.1079	0.3551	0.4067	0.6022
14	0.5818	0.5818	0.4818	0.3182	0.0000	0.0273	0.0000	0.0273	0.0182	0.4455
15	0.2727	0.4000	0.7818	0.5727	0.2636	0.3727	0.0545	0.4091	0.4455	0.7818
16	0.0748	0.0748	0.1121	0.1511	0.0016	0.0016	0.0000	0.0763	0.0000	0.0000
17	0.0000	0.0000	0.3878	0.2653	0.0000	0.0000	0.0000	0.0000	0.0000	0.1243
18	0.0000	0.0000	0.0187	0.0327	0.0000	0.0000	0.0000	0.0000	0.0000	0.0016
19	0.0000	0.0000	0.1651	0.1651	0.0000	0.0000	0.0000	0.0000	0.0148	0.0019
20	0.0748	0.0249	0.0514	0.0125	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
21	0.0286	0.0286	0.6143	0.4714	0.0143	0.0000	0.0143	0.0143	0.2286	0.2286
22	0.2857	0.2571	0.5857	0.4714	0.0000	0.0143	0.0000	0.0000	0.0000	0.0143
23	0.3077	0.3077	0.6410	0.4103	0.0000	0.0000	0.0000	0.0000	0.0513	0.0000
24	0.3645	0.3505	0.5062	0.6542	0.1433	0.0997	0.0810	0.2056	0.2570	0.8832
25	0.1036	0.1036	0.4966	0.3563	0.0108	0.0340	0.0050	0.0895	0.0215	0.0000
26	0.4714	0.4714	0.7286	0.7286	0.0000	0.0000	0.0000	0.0286	0.0286	0.0143
27	0.1509	0.1509	0.4088	0.4969	0.0818	0.1006	0.0503	0.1509	0.0000	0.0566
28	0.4717	0.4717	0.7799	0.5094	0.1195	0.1509	0.0189	0.3270	0.3208	0.3208
29	0.0714	0.0286	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
30	0.3687	0.3687	0.5622	0.3088	0.0046	0.0369	0.0000	0.1106	0.4009	0.4516
31	0.2556	0.0444	0.3611	0.1389	0.0000	0.0000	0.0000	0.0750	0.3028	0.3083
32	0.2222	0.1889	0.1306	0.0000	0.0000	0.0000	0.0000	0.0000	0.3083	0.4194
33	0.4714	0.4714	0.6571	0.3429	0.0000	0.0000	0.0000	0.1714	0.3286	0.3714
34	0.3286	0.3286	0.4714	0.4714	0.0000	0.0000	0.0000	0.4714	0.2429	0.7286
35	0.4714	0.4714	0.5571	0.5000	0.0000	0.0857	0.0143	0.0143	0.2000	0.5000
36	0.4528	0.4528	0.8176	0.5094	0.1006	0.0252	0.1321	0.6038	0.0000	0.4969
37	0.4286	0.6429	0.2286	0.1857	0.0429	0.0429	0.0286	0.7571	0.2714	0.3429
38	0.4182	0.4182	0.7364	0.5909	0.0000	0.0000	0.0000	0.0000	0.2091	0.4636
39	0.4906	0.4906	0.8176	0.6604	0.3082	0.2138	0.0126	0.7170	0.4969	0.6415
40	0.7286	0.7286	0.7286	0.7286	0.0000	0.0286	0.0000	0.0000	0.0000	0.4857
41	0.5727	0.5727	0.7818	0.6818	0.0091	0.0455	0.0000	0.2000	0.7818	0.7818
42	0.4714	0.4714	0.7286	0.5286	0.0000	0.0000	0.0000	0.1571	0.5857	0.1429
43	0.5571	0.5429	0.7286	0.4714	0.0286	0.3429	0.0429	0.0429	0.2286	0.1429
44	0.7818	0.7818	0.5909	0.7818	0.2545	0.2545	0.0000	0.3909	0.7818	0.7818
45	0.7286	0.7286	0.7286	0.6571	0.0286	0.2000	0.0286	0.0857	0.4857	0.3143
46	0.6162	0.5563	0.8627	0.3838	0.0775	0.4472	0.1197	0.1127	0.3908	0.6268
47	0.8176	0.8176	0.8176	0.6415	0.1824	0.2516	0.1572	0.7799	0.8176	0.6604
48	0.6410	0.6410	0.6410	0.6410	0.0000	0.0000	0.0000	0.0000	0.1282	0.3590
49	0.4906	0.4906	0.8176	0.6415	0.2201	0.2327	0.2453	0.4717	0.6415	0.6415
50	0.6667	0.6667	0.8778	0.7778	0.0750	0.1806	0.0139	0.3889	0.3972	0.3056
ANMRR	0.3410	0.3368	0.5622	0.4173	0.0426	0.0696	0.0234	0.1817	0.2350	0.3281

Table 6.6: Results of descriptors using MPEG-7 CCD

Query	DCD	CLD	EHD	DCD+CLD+EHD
1	0.7286	0.5429	0.5286	0.2286
2	0.4528	0.5472	0.4717	0.3585
3	0.0517	0.0472	0.0045	0.0000
4	0.1218	0.0729	0.3471	0.0000
5	0.1920	0.0019	0.0160	0.0012
6	0.1194	0.1500	0.2861	0.0111
7	0.0599	0.2500	0.4401	0.0106
8	0.1073	0.1491	0.1693	0.0000
9	0.0000	0.0000	0.1290	0.1290
10	0.0377	0.0063	0.1698	0.0000
11	0.0727	0.0909	0.0273	0.0091
12	0.1091	0.1000	0.3727	0.3091
13	0.6584	0.4067	0.6022	0.0517
14	0.5818	0.0182	0.4455	0.0545
15	0.4000	0.4455	0.7818	0.2909
16	0.0748	0.0000	0.0000	0.0031
17	0.0000	0.0000	0.1243	0.0000
18	0.0000	0.0000	0.0016	0.0000
19	0.0000	0.0148	0.0019	0.0000
20	0.0249	0.0000	0.0000	0.0000
21	0.0286	0.2286	0.2286	0.0714
22	0.2571	0.0000	0.0143	0.0000
23	0.3077	0.0513	0.0000	0.3077
24	0.3505	0.2570	0.8832	0.3380
25	0.1036	0.0215	0.0000	0.0224
26	0.4714	0.0286	0.0143	0.0000
27	0.1509	0.0000	0.0566	0.1509
28	0.4717	0.3208	0.3208	0.3082
29	0.0286	0.0000	0.0000	0.0000
30	0.3687	0.4009	0.4516	0.2719
31	0.0444	0.3028	0.3083	0.0083
32	0.1889	0.3083	0.4194	0.3083
33	0.4714	0.3286	0.3714	0.0000
34	0.3286	0.2429	0.7286	0.2429
35	0.4714	0.2000	0.5000	0.0286
36	0.4528	0.0000	0.4969	0.1887
37	0.6429	0.2714	0.3429	0.0000
38	0.4182	0.2091	0.4636	0.3727
39	0.4906	0.4969	0.6415	0.6415
40	0.7286	0.0000	0.4857	0.4714
41	0.5727	0.7818	0.7818	0.7818
42	0.4714	0.5857	0.1429	0.2429
43	0.5429	0.2286	0.1429	0.1286
44	0.7818	0.7818	0.7818	0.6636
45	0.7286	0.4857	0.3143	0.3286
46	0.5563	0.3908	0.6268	0.3979
47	0.8176	0.8176	0.6604	0.6415
48	0.6410	0.1282	0.3590	0.0000
49	0.4906	0.6415	0.6415	0.6415
50	0.6667	0.3972	0.3056	0.3417
ANMRR	0.3368	0.2350	0.3281	0.1872

Table 6.7: Comparison between fusion and isolated descriptors using MPEG-7 CCD

Chapter 7

Conclusions and Future Work

In this project, a search engine for image retrieval has been presented. This CBIR system uses a set of low-level features called visual descriptors that have been defined by MPEG-7. Thus, the implemented tool is based on a standard, an effort to increase its reliability. Only in a few cases some variations have been proposed in order to improve the results. Sometimes, particularly for the extraction of these descriptors for regions, these variations have been proposed because of a lack of some details in the standard.

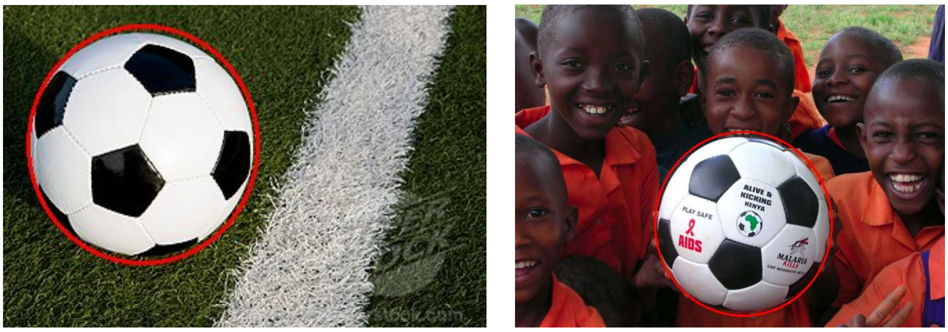
These visual descriptors have been classified depending on which feature describes, such as color, texture, and shape. Although all the descriptors have been implemented both for images and regions, only the ones that are extracted for the whole image have been used. That is the reason why the Contour Shape Descriptor has not been considered in spite of its implementation. Thus, any shape descriptor does not make sense for image retrieval systems.

After implementing all these descriptors, a dissimilarity measure has to be chosen for each one. Given two images from which a visual descriptor has been obtained, this distance gives us an idea of how similar these images are according to the extracted feature. Thus, these dissimilarity measures allow to obtain a quantitative result that can be used for sorting out the target images objectively. Various distances have been proposed and analyzed for each descriptor.

Once the target images are sorted out for each descriptor, a method that allows to fuse all the results obtained in order to rank these images by similarity is applied. This method takes advantage of the idea that using a set of descriptors leads to some better results than the ones obtained using an isolated descriptor. As it has shown in Chapter 6, this idea is right only if the used descriptors have similar behaviors. This fusion method consists in weighting each rank list obtained from each descriptor in order to create a new rank list gathering all the descriptors.

According to the results obtained, we draw the conclusion that among all the implemented descriptors the Color Structure Descriptor gives by far the best performance. Another conclusion to which come is that fusioning the results obtained by Color Layout, Dominant Color and Texture Edge Histogram descriptors improves the performance. Except for texture descriptors, we have used the same database as MPEG-7 so that we can compare the results obtained.

This project opens the door to new future work lines. The main work line consists in developing a region-based search engine. For this system, we will incorporate shape descriptors, as well as the whole set of descriptors used in image retrieval, which have been already implemented to be extracted from arbitrary shapes. In this new approach, the Binary Partition Tree will play an important role because each of its node will represent a region to be considered. Figure 7.1 depicts an example of region-base query by example. In this, we show a picture that we would expect to be retrieved when a soccer ball is used as a query.

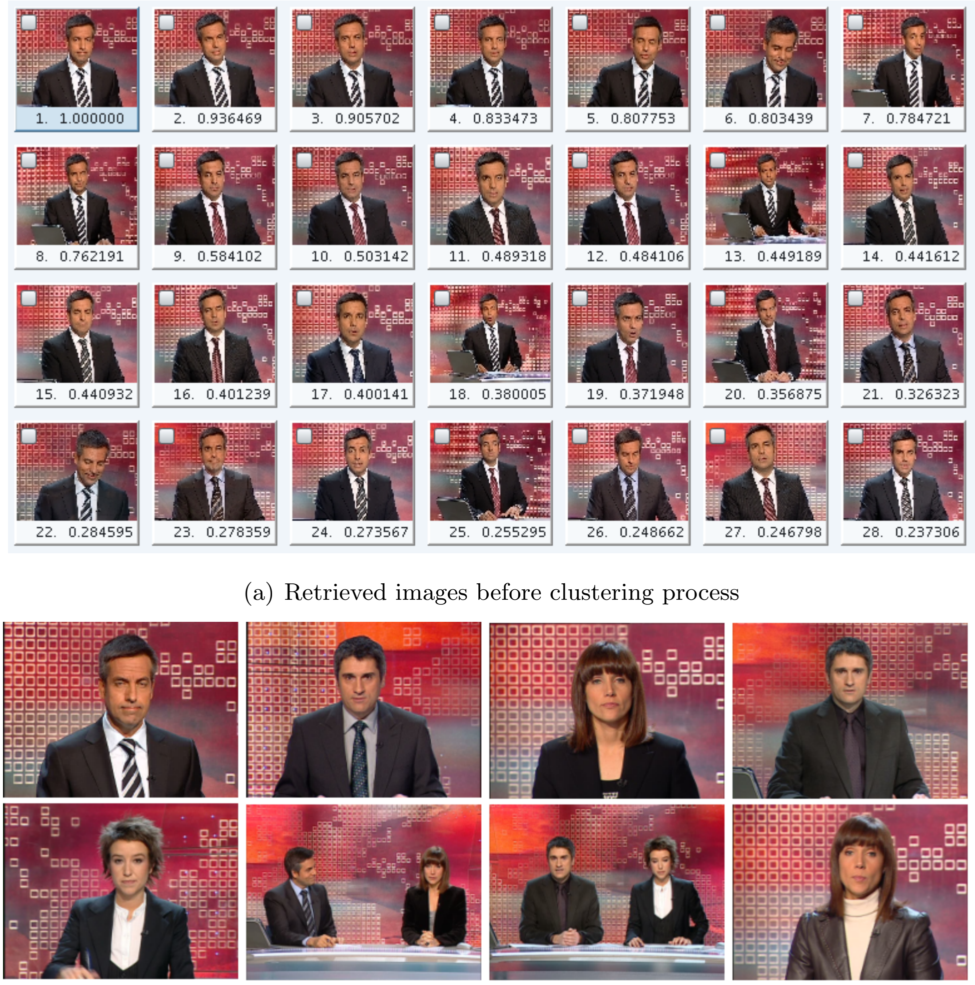


(a) query region

(b) retrieved region

Figure 7.1: Example of region-based query by example. Figure (b) represents an image we expect to be retrieved when the soccer ball of Figure (a) is used as a query.

Another future work resulting from this project is the clustering process. This consists in putting the retrieved images into groups. A typical case is when the images of a database are frames of video sequences. Then, it could be interesting to show only a retrieved image for each video sequence instead of a set of images, which are very similar because of their temporal redundancy. In general, this clustering process will not only rely on temporal redundancy. Thus, for example, if we are searching a news anchorperson without clustering the results, the retrieved top images will show the same anchorperson in the same scenario despite belonging to different video sequences. This example is illustrated in Figure 7.2.



(a) Retrieved images before clustering process

(b) Retrieved images after clustering process

Figure 7.2: Example of clustering process. Figure (a) depicts the retrieved images before the clustering process, whereas Figure (b) shows which could be the retrieved images after the clustering process.

Relevance Feedback (RF) techniques are a way to improve the results obtained. These techniques consist in involving the human factor in the search engine. Thus, once the retrieved images have been shown, the user have to decide which results are considered relevant. Then, the retrieval system takes advantage of this information given by the user and tries to improve its results by using all the features extracted from these relevant images. Figure 7.3 shows an example in which the user is searching a close-up of a football player and has marked the images considered as relevant.

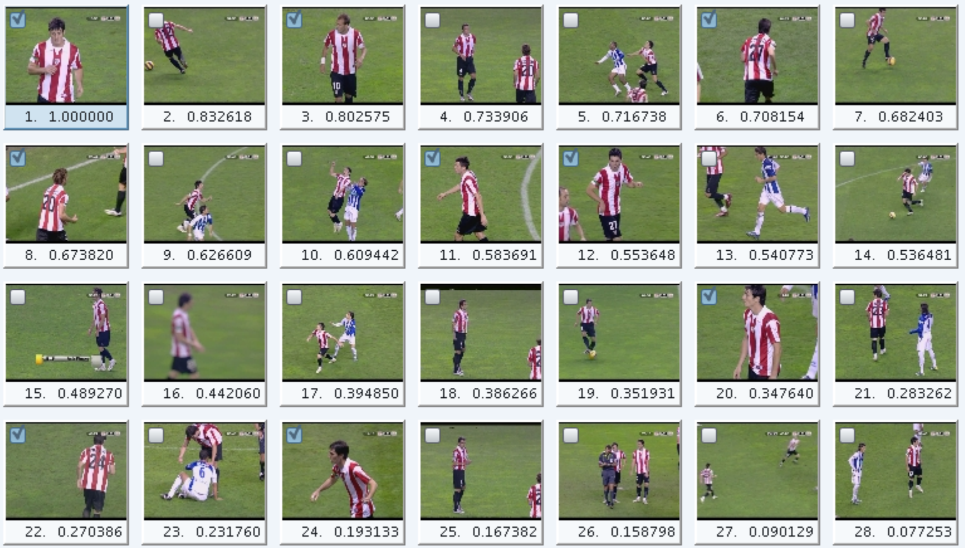


Figure 7.3: Example of relevance feedback techniques. The user is looking for a close-up picture of a football player and have considered the images marked by a tick as the relevant ones.

Another research line consists in studying other ways to combine the different results obtained by each visual descriptor. In this project has been decided not to take into account the distances computed any more and consider only the rank lists, which are weighted to create the resulting rank. Another method that could be researched is based on normalizing each distance for each descriptor and weighting these values in order to obtain a global distance. The main drawback of this technique is that it requires a previous study of the statistical distribution of the distance values obtained for each descriptor in order to normalize each distance before combining them.

These three latter work lines presented will be useful for both image and region retrieval. The first one, which consists in a region-based query by example, will be the basis of the research I expect to conduct during my PhD thesis.

Appendix A

Binary Partition Tree

As it is defined in [Ost02], the Binary Partition Tree (BPT) is a structured representation of the regions that can be obtained from an initial partition. In other words, it is a structured representation of a set of hierarchical partitions in which the nest level of detail is given by the initial partition. The leaves of the tree represent regions that belong to this initial partition. The remaining nodes of the tree are associated to regions that represent the union of two children regions. The root node usually represents the entire image support. This representation should be considered as a compromise between representation accuracy and processing efficiency. Indeed, all possible mergings of regions belonging to the initial partition (described by the RAG of the initial partition) are not represented in the tree. Only the most "likely" or "useful" merging steps are represented in the BPT. The connectivity encoded in the tree structure is binary in the sense that a region is explicitly connected to its sibling (since their union is a connected component represented by the parent), but the remaining connections between regions of the original partition are not represented in the tree. Therefore, the tree encodes only part of the neighborhood relationships between the regions of the initial partition.

The Binary Partition Tree should be created in such a way that the most "interesting" or "useful" regions are represented. This issue can be application dependent. However, a possible solution, suitable for a large number of cases, is to create the tree by keeping track of the merging steps performed by a segmentation algorithm based on region merging. This information is called the *merging sequence*. Starting from an initial partition which can be the partition of at zones or any other pre-computed partition, the algorithm merges neighboring regions following a homogeneity criterion until a single region is obtained.

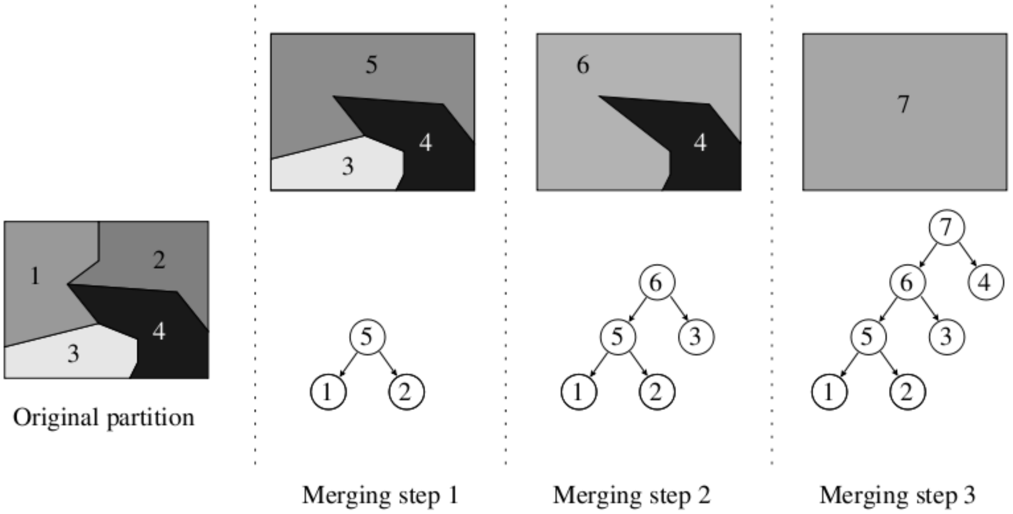


Figure A.1: Example of Binary Partition Tree creation extracted from [Ost02]

An example is shown in Figure A.1. The original partition involves four regions. The algorithm merges the four regions in three steps. In the rst step, the pair of most similar regions, 1 and 2, are merged to create region 5. This is indicated in the Binary Partition Tree with a node whose label is 5 and that has to children nodes, 1 and 2. Then, region 5 is merged with region 3 to create region 6. Finally, region 6 is merged with region 4 and this crates region 7 corresponding to the region of support of the whole image. In this example, the merging sequence is: $(1,2)\|(5,3)\|(6,4)$. This merging sequence progressively denes the Binary Partition Tree as shown in Fig. 4.1. In this case the initial partition is made up of 4 regions and thus, the number of nodes of the tree is $4+(4-1)=7$.

In a more general case, we may start creating the tree from an initial partition P made of N_P regions. The number of mergings that are needed to obtain one region is N_P-1. Therefore, the number of nodes of the Binary Partition Tree is thus $2N_P-1$.

Appendix B

Color Spaces

There are many color spaces designed for different systems and standards, but most of them can be obtained by a simple transformation. Next, a brief description is given for the color spaces used by some visual descriptors:

- The *RGB* color space is one of the most popular models. This space is defined as the unit cube in the Cartesian coordinate system which has Red (R), Green (G) and Blue (B) additive primaries as a basis. These colors are added together in various ways to reproduce a broad array of colors. Each of the three primaries is called a component of that color, and each of them can have an arbitrary intensity, from fully off to fully on, in the mixture. Zero intensity for each component gives the darkest color, and full intensity of each gives a white. When the intensities for all the components are the same, the result is a shade of gray, darker or lighter depending on the intensity. When the intensities are different, the result is a colorized hue, more or less saturated depending on the difference of the strongest and weakest of the intensities of the primary colors employed.

- *YCbCr* is another color space where the component Y represents the luma, i.e. the brightness, and Cb and Cr are the blue difference (B-Y) and the red difference (R-Y) components, respectively. One advantage of the YUV color format is based on the characteristics of the human visual perception: Since the human eye is much more sensitive for brightness information compared to color information, we can give less importance to the chrominance information in some dissimilarity measures that are computed in this color space, such as Color Layout Descriptor.

- The *Monochrome* color space uses only the luma component (Y) of the YCbCr color space.

- The *HSV* color space is developed to provide an intuitive representation of color and to approximate the way in which humans perceive and manipulate color. Hue (H) specifies one color family from another, as red from yellow, green, blue or purple. Saturation (S) specifies how pure a color is and Value (V) specifies how bright or dark a color is. The HSV color model can be seen as a double-cone. The angle around the central axis represents the hue, the distance to the axis represents the saturation and position along the central axis represents the value or luminance.

- The *HMMD* (Hue-Max-Min-Diff) color space is closer to a perceptually uniform color space. The Hue has the same meaning as in the HSV color space. Max and Min components are the maximum and minimum among the R, G, B values, respectively. The Diff component is defined as the difference between max and min. Even though the four components are identified in the name of the color space, one more component, Sum, can be defined as the average of Min and Max components. Only three of the five components are sufficient to describe the HMMD color space.

More information about these color spaces in the context of image indexing and retrieval can be found in [MPS02].

Appendix C

List of Ground Truth Sets for MPEG-7 CCD

Table C.1: MPEG-7 CCD Ground Truth Sets

Query Name	Query Image	Ground Truth Images
1.Flower garden	img00587_add3.jpg	img00587_add3.jpg, img00585_add3.jpg, img00586_add3.jpg, img00588_add3.jpg
2.Rock and sky	img0066d_s1.jpg	img0066d_s1.jpg, img0063d_s1.jpg, img0064d_s1.jpg, img0065d_s1.jpg, img0067d_s1.jpg, img0054d_s1.jpg
3. NEWS anchor	jornal64742_add2.jpg	jornal64742_add2.jpg, JORNA∼18_add2.jpg, JORNA∼19_add2.jpg, JORNA∼20_add2.jpg, JORNA∼21_add2.jpg, jornal48182_add2.jpg, jornal52142_add2.jpg, jornal67982_add2.jpg, jornal68162_add2.jpg, jornal71042_add2.jpg

Table C.1

Query Name	**Query Image**	**Ground Truth Images**
4. Walking people	NEWS1∼46_add2.jpg	NEWS1∼46_add2.jpg, NEWS1∼53_add2.jpg, i6b_add1.jpg, NEWS1∼42_add2.jpg, NEWS1∼43_add2.jpg, NEWS1∼44_add2.jpg, NEWS1∼45_add2.jpg, NEWS1∼47_add2.jpg, NEWS1∼48_add2.jpg, NEWS1∼49_add2.jpg, NEWS1∼50_add2.jpg, NEWS1∼51_add2.jpg, NEWS1∼52_add2.jpg, NEWS1∼54_add2.jpg, NEWS1∼55_add2.jpg, NEWS1∼56_add2.jpg, NEWS1∼57_add2.jpg
5. Baldhead man walking and talking with persons	NEWS1650_add2.jpg	NEWS1650_add2.jpg, news1614_add2.jpg, i2b_add1.jpg, news1552_add2.jpg, news1561_add2.jpg, news1570_add2.jpg, NEWS1578_add2.jpg, NEWS1587_add2.jpg, NEWS1596_add2.jpg, NEWS1605_add2.jpg, NEWS1623_add2.jpg, NEWS1632_add2.jpg, NEWS1641_add2.jpg, NEWS1659_add2.jpg, NEWS1668_add2.jpg, NEWS1677_add2.jpg, NEWS1686_add2.jpg, NEWS1695_add2.jpg, NEWS1704_add2.jpg, NEWS1713_add2.jpg, NEWS1722_add2.jpg, NEWS1731_add2.jpg, NEWS1740_add2.jpg, NEWS1749_add2.jpg,
6.Sports reporters in the rain	SPOR∼214_add2.jpg	SPOR∼214_add2.jpg, i48c_add1.jpg, SPOR∼210_add2.jpg, SPOR∼211_add2.jpg, SPOR∼212_add2.jpg, SPOR∼213_add2.jpg, SPOR∼215_add2.jpg, SPOR∼216_add2.jpg, SPOR∼217_add2.jpg
7.Congress	JORNA∼89_add2.jpg	JORNA∼89_add2.jpg, i8a_add1.jpg, JORNA∼85_add2.jpg, JORNA∼86_add2.jpg, JORNA∼87_add2.jpg, JORNA∼88_add2.jpg, JORNA∼90_add2.jpg, JORNA∼91_add2.jpg

Table C.1

Query Name	**Query Image**	**Ground Truth Images**
8.Baldheaded man representing	JORNA∼54_add2.jpg	JORNA∼54_add2.jpg, JORNA∼43_add2.jpg, JORNA∼44_add2.jpg, JORNA∼45_add2.jpg, JORNA∼46_add2.jpg, JORNA∼47_add2.jpg, JORNA∼48_add2.jpg, JORNA∼49_add2.jpg, JORNA∼50_add2.jpg, JORNA∼51_add2.jpg, JORNA∼52_add2.jpg, JORNA∼53_add2.jpg, JORNA∼55_add2.jpg, JORNA∼56_add2.jpg, JORNA∼57_add2.jpg, JORNA∼58_add2.jpg, JORNA∼59_add2.jpg, JORNA∼60_add2.jpg, JORNA∼61_add2.jpg, JORNA∼62_add2.jpg, JORNA∼63_add2.jpg, JORNA∼64_add2.jpg, JORNA∼65_add2.jpg, JORNA∼66_add2.jpg, JORNA∼67_add2.jpg, JORNA∼68_add2.jpg, JORNA∼69_add2.jpg, JORNA∼70_add2.jpg, JORNA∼71_add2.jpg, jornal1022_add2.jpg, jornal662_add2.jpg, jornal842_add2.jpg
9.Castle	CULTUR∼7_add2.jpg	CULTUR∼7_add2.jpg, CULTUR∼4_add2.jpg, CULTUR∼5_add2.jpg, CULTUR∼6_add2.jpg, CULTUR∼8_add2.jpg, CULTUR∼9_add2.jpg, CULTU∼10_add2.jpg
10.Black clothes lady on the blue mat	i140n_add1.jpg	i140n_add1.jpg, i143n_add1.jpg, i131n_add1.jpg, i135n_add1.jpg, i139n_add1.jpg, i150n_add1.jpg
11.Singer with studio lights	D03_add4.jpg	D03_add4.jpg, D04_add4.jpg, D05_add4.jpg, D06_add4.jpg, D09_add4.jpg
12.Strange hair	img0020b_s1.jpg	img0020b_s1.jpg, img0021b_s1.jpg, img0022b_s1.jpg, img0023b_s1.jpg, img0024b_s1.jpg

Table C.1

Query Name	Query Image	Ground Truth Images
13.Leather jacket people	img0027b_s1.jpg	img0027b_s1.jpg, img0026b_s1.jpg, img0028b_s1.jpg, img0029b_s1.jpg, img0030b_s1.jpg, img0031b_s1.jpg, img0032b_s1.jpg, img0033b_s1.jpg, img0034b_s1.jpg, img0035b_s1.jpg
14.Man with placard	img0064a_s1.jpg	img0064a_s1.jpg, img0065a_s1.jpg, img0066a_s1.jpg, img0067a_s1.jpg, img0069a_s1.jpg
15.People on the red	img01271_s3.jpg	img01271_s3.jpg, img01264_s3.jpg, img01266_s3.jpg, img01267_s3.jpg, img01268_s3.jpg
16.Snake	i0312_add5.jpg	i0312_add5.jpg, i0313_add5.jpg, i0314_add5.jpg, i0315_add5.jpg, i0316_add5.jpg, i0317_add5.jpg, i0318_add5.jpg, i0319_add5.jpg, i0320_add5.jpg, i0321_add5.jpg, i0322_add5.jpg, i36m_add1.jpg
17.Fish	i0323_add5.jpg	i0323_add5.jpg, i0324_add5.jpg, i0325_add5.jpg, i0326_add5.jpg, i0327_add5.jpg, i0328_add5.jpg, i0329_add5.jpg, i0330_add5.jpg, i0331_add5.jpg, i0332_add5.jpg, i0333_add5.jpg

Table C.1

Query Name	**Query Image**	**Ground Truth Images**
18.Tapirs	i0334_add5.jpg	i0334_add5.jpg, i0335_add5.jpg, i0336_add5.jpg, i0337_add5.jpg, i0338_add5.jpg, i0339_add5.jpg, i0340_add5.jpg, i0341_add5.jpg, i0342_add5.jpg, i0343_add5.jpg, i0344_add5.jpg, i46m_add1.jpg
19.Butterfly	i0356_add5.jpg	i0356_add5.jpg, i0357_add5.jpg, i0358_add5.jpg, i0359_add5.jpg, i0360_add5.jpg, i0361_add5.jpg, i0362_add5.jpg, i0363_add5.jpg, i0364_add5.jpg, i0365_add5.jpg, i0366_add5.jpg
20.Small monkey	i0367_add5.jpg	i0367_add5.jpg, i0368_add5.jpg, i0369_add5.jpg, i0370_add5.jpg, i0371_add5.jpg, i0372_add5.jpg, i0373_add5.jpg, i0374_add5.jpg, i0375_add5.jpg, i0376_add5.jpg, i0377_add5.jpg, i44m_add1.jpg
21.Landscape Image 1	img0002a_s1.jpg	img0002a_s1.jpg, img0001a_s1.jpg, img0091a_s1.jpg, img0003a_s1.jpg
22.Landscape Image 2	img02025_s3.jpg	img02025_s3.jpg, img02021_s3.jpg. img02013_s3.jpg, img02019_s3.jpg
23.Landscape Image 3	img02029_s3.jpg	img02029_s3.jpg, img02028_s3.jpg, img02023_s3.jpg

Table C.1

Query Name	Query Image	Ground Truth Images
24.Indoor Image	img01381_s3.jpg	img01381_s3.jpg, img01382_s3.jpg, img01373_s3.jpg, img01375_s3.jpg, img01358_s3.jpg, img01364_s3.jpg, img01380_s3.jpg, img01442_s3.jpg, img01757_s3.jpg, img01768_s3.jpg, img01926_s3.jpg, img01933_s3.jpg
25.Anchorperson	jornal28202_add2.jpg	jornal28202_add2.jpg, jornal1742_add2.jpg, i11a_add1.jpg, jornal1922_add2.jpg, jornal2102_add2.jpg, jornal10022_add2.jpg, jornal16142_add2.jpg, jornal16322_add2.jpg, jornal28382_add2.jpg, jornal32162_add2.jpg, jornal32342_add2.jpg, jornal32522_add2.jpg, jornal35222_add2.jpg, jornal35402_add2.jpg, jornal54122_add2.jpg, jornal54302_add2.jpg, jornal57902_add2.jpg
26. Quiz scene	i0121_add5.jpg	i0121_add5.jpg, i0123_add5.jpg, i26e_add1.jpg, i0131_add5.jpg
27.Speaker	img01179_s3.jpg	img01179_s3.jpg, img01178_s3.jpg, img01180_s3.jpg, img01184_s3.jpg, img01186_s3.jpg, img01183_s3.jpg
28.Man and horse	CULTU∼47_add2.jpg	CULTU∼47_add2.jpg, CULTU∼48_add2.jpg, CULTU∼49_add2.jpg, CULTU∼50_add2.jpg, CULTU∼56_add2.jpg, CULTU∼57_add2.jpg
29.Space earth	NEWS147_add2.jpg	NEWS147_add2.jpg, NEWS138_add2.jpg, NEWS156_add2.jpg, NEWS165_add2.jpg

Table C.1

Query Name	Query Image	Ground Truth Images
30.Fountain	img00133_add3.jpg	img00133_add3.jpg, img00134_add3.jpg, img00136_add3.jpg, img00137_add3.jpg, img00130_add3.jpg, img00131_add3.jpg, img00135_add3.jpg
31.Graphics before NEWS	jornal71222_add2.jpg	jornal71222_add2.jpg, JORNAL∼8_add2.jpg, JORNAL∼9_add2.jpg, JORNA∼10_add2.jpg, JORNA∼11_add2.jpg, JORNA∼12_add2.jpg, JORNA∼13_add2.jpg, JORNA∼14_add2.jpg, JORNA∼15_add2.jpg
32.Ron Reagan	img00438_s3.jpg	img00438_s3.jpg, img00439_s3.jpg, img00440_s3.jpg, img00441_s3.jpg, img00442_s3.jpg, img00444_s3.jpg, img00445_s3.jpg, img00446_s3.jpg, img00447_s3.jpg
33.Basketball GAME overlay	GAME12∼9_add2.jpg	GAME12∼9_add2.jpg, GAME12∼3_add2.jpg, GAME12∼2_add2.jpg, GAME1∼12_add2.jpg
34.Glass roof	img00121_add3.jpg	img00121_add3.jpg, img00122_add3.jpg, img00123_add3.jpg, img00126_add3.jpg
35. Snow clad mountain	img01115_add3.jpg	img01115_add3.jpg, img01116_add3.jpg, img01117_add3.jpg, img01114_add3.jpg
36.Outdoor/boats	img00071_add3.jpg	img00071_add3.jpg, img00070_add3.jpg, img00086_add3.jpg, img00085_add3.jpg, img00084_add3.jpg, img00058_add3.jpg
37.By the water	img00537_add3.jpg	img00537_add3.jpg, img00538_add3.jpg, img00616_add3.jpg, img00625_add3.jpg

Table C.1

Query Name	**Query Image**	**Ground Truth Images**
38.Couple	i0198_add5.jpg	i0198_add5.jpg, i0197_add5.jpg, i0192_add5.jpg, i0191_add5.jpg, i0190_add5.jpg
39.Shop	img01023_add3.jpg	img01023_add3.jpg, img01021_add3.jpg, img01022_add3.jpg, img01024_add3.jpg, img01026_add3.jpg, img01027_add3.jpg
40.Flower(indoor)	img00158_add3.jpg	img00158_add3.jpg, img00161_add3.jpg, img00159_add3.jpg, img00160_add3.jpg
41.Playing on the street	img00283_add3.jpg	img00283_add3.jpg, img00276_add3.jpg, img00281_add3.jpg, img00282_add3.jpg, img00350_add3.jpg
42.Road with trees/grass	img00442_add3.jpg	img00442_add3.jpg, img00440_add3.jpg, img00444_add3.jpg, img00445_add3.jpg
43.Children/rock/grass	img00867_add3.jpg	img00867_add3.jpg, img00864_add3.jpg, img00866_add3.jpg, img00865_add3.jpg
44.Asian building	img01094_add3.jpg	img01094_add3.jpg, img01093_add3.jpg, img01090_add3.jpg, img01099_add3.jpg, img01100_add3.jpg
45.Containers	img00097_s3.jpg	img00097_s3.jpg, img00090_s3.jpg, img00089_s3.jpg, img00099_s3.jpg
46.Sunset over lake	img00136_s3.jpg	img00136_s3.jpg, img00135_s3.jpg, img00138_s3.jpg, img00204_s3.jpg, img00226_s3.jpg, img00720_s3.jpg, img01155_s3.jpg, img00273_s3.jpg

Table C.1

Query Name	**Query Image**	**Ground Truth Images**
47.Big pipes	img01903_s3.jpg	img01903_s3.jpg, img01893_s3.jpg, img01909_s3.jpg, img01911_s3.jpg, img01912_s3.jpg, img01922_s3.jpg
48.Man with sunglasses in white shirt	img0016b_s1.jpg	img0016b_s1.jpg, img0015b_s1.jpg, img0017b_s1.jpg
49.Wooden shack	img0078d_s1.jpg	img0078d_s1.jpg, img0014d_s1.jpg, img0077d_s1.jpg, img0080d_s1.jpg, img0081d_s1.jpg, img0083d_s1.jpg
50.Ruins	img0023d_s1.jpg	img0023d_s1.jpg, img0018d_s1.jpg, img0019d_s1.jpg, img0020d_s1.jpg, img0022d_s1.jpg, img0024d_s1.jpg, img0025d_s1.jpg, img0026d_s1.jpg, img0027d_s1.jpg

Bibliography

[BFG96] J. R. Bach, C. Fuller, and A. Gupta, *Virage image search engine: an open framework for image management*, Storage and Retrieval for Still Image and Video Databases IV **2670** (1996), no. 1, 76–87.

[CER99] *Core Experiment Results for Spatial Intensity Descriptor*, ISO/IEC/JTC1/SC29/WG11, Dec.1999, MPEG Document M5374.

[EXC98] *Excalibur Visual Retrievalware*, web page: `http://www.excalib.com/products/vrw/vrw.html`, 1998.

[FSN$^+$97] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, *Query by image and video content: the QBIC system*, Intelligent multimedia information retrieval, MIT Press, Cambridge, MA, USA, 1997, pp. 7–22.

[GG93] A. Gersho and R. M. Gray, *Vector quantization and signal compression*, Kluwer Academic Publishers, Norwell, Mass, 1993.

[HK92] K. Hirata and T. Kato, *Query by Visual Example - Content based Image Retrieval*, EDBT '92: Proceedings of the 3rd International Conference on Extending Database Technology (London, UK), Springer-Verlag, 1992, pp. 56–71.

[HMR96] T. S. Huang, S. Mehrotra, and K. Ramachandran, *Multimedia Analysis and Retrieval System (MARS) project*, in Proc. of 33rd Annual Clinic on Library Application of Data Processing - Digital Image Access and Retrieval, 1996.

[IMG] *ImgSeek*, web page: `http://www.imgseek.net/`.

[JFS95] C. E. Jacobs, A. Finkelstein, and D. H. Salesin, *Fast multiresolution image querying*, SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (New York, NY, USA), ACM, 1995, pp. 277–286.

[Kuh55] H. W. Kuhn, *The Hungarian method for the assignment problem*, Naval Research Logistic Quarterly **2** (1955), 83–97.

[KY01] E. Kasutani and A. Yamada, *The MPEG-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval*, Proc. International Conference on Image Processing, vol. 1, 2001, pp. 674–677.

[MAK96] F. Mokhtarian, S. Abbasi, and J. Kittler, *Robust and efficient shape indexing through curvature scale space*, In Proceedings of British Machine Vision Conference, 1996, pp. 53–62.

[MM97] W.Y. Ma and B.S. Manjunath, *Netra: a toolbox for navigating large image databases*, International Conference on Image Processing, vol. 1, IEEE Computer Society, Los Alamitos, CA, USA, 1997, p. 568.

[MP3] *Multimedia Content Description Interface - Part3: Visual*, ISO/IEC 15938-3:2001, Version 1.

[MP8] *Multimedia Content Description Interface - Part8: Extraction and Use of MPEG-7 Descriptions*, ISO/IEC 15938-8:2001, Version 1.

[MPS02] B. S. Manjunath, P.Salembier, and T. Sikora, *Introduction to MPEG-7, Multimedia Content Description Interface*, John Wiley and Sons, Ltd., Jun 2002.

[MUL] *Multicolr Search Lab*, web page: `http://labs.ideeinc.com/multicolr/`.

[MvBE01] D.S. Messing, P. van Beek, and J.H. Errico, *The MPEG-7 colour structure descriptor: image description using colour and local spatial information*, Proc. International Conference on Image Processing, vol. 1, 2001, pp. 670–673.

[NBE$^+$93] C. W. Niblack, R. Barber, W. Equitz, M. D. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, *QBIC project: querying images by content, using color, texture, and shape*, Storage and Retrieval for Image and Video Databases, vol. 1908, SPIE, 1993, pp. 173–187.

[NMH02] M. Nakazato, L. Manola, and T. S. Huang, *ImageGrouper: Search, annotate and organize images by groups*, in: Proceedings of Visual Information Systems Conference, 2002, HSinChu, Taiwan, pp. 129–142.

[NRM$^+$00] P. Ndjiki-Nya, J. Restat, T. Meiers, J. R. Ohm, A. Seyferth, and R. Sniehotta, *Subjective evaluation of the mpeg-7 retrieval accuracy measure (ANMRR)*, ISO/IEC/JTC1/WG11, May.2000.

[Ost02] L.G. Ostermann, *Hierarchical Region Based Processing of Images and Video Sequences: Application to Filtering, Segmentation and Information Retrieval*, vol. 1, Department of Signal Theory and Communications, UPC, Barcelona, Spain, April 2002, pp. 53–70.

[PJW00] D. K. Park, Y. S. Jeon, and C. S. Won, *Efficient use of local edge histogram descriptor*, ACM, 2000, pp. 51–54.

[PM08] J. Pont and F. Marquès, *On the definition of a similarity measure between patterns*, TSC(UPC) - Image and Video Processing Group, 2008, Internal Document.

[PPS95] A. Pentland, R. W. Picard, and S. Sclaroff, *Photobook: Content-Based Manipulation of Image Databases*, 1995.

[PW04] L. M. Po and K. M. Wong, *A new palette histogram similarity measure for MPEG-7 dominant color descriptor*, International Conference on Image Processing, vol. 3, Oct. 2004, pp. 1533–1536.

[RHC99] Y. Rui, T. S. Huang, and S. Chang, *Image Retrieval: Current Techniques, Promising Directions and Open Issues*, Journal of Visual Communication and Image Representation, vol. 10, 1999, pp. 39–62.

[RTG00] Y. Rubner, C. Tomasi, and L. J. Guibas, *The earth mover's distance as a metric for image retrieval*, International Journal of Computer Vision, vol. 40, Nov 2000, pp. 99–121.

[SC96] J. R. Smith and S. Chang, *VisualSEEk: a fully automated content-based image query system*, MULTIMEDIA '96: Proceedings of the fourth ACM international conference on Multimedia, 1996, New York, USA, pp. 87–98.

[SC97] J. R. Smith and S. Chang, *Visually Searching the Web for Content*, vol. 4, IEEE Computer Society Press, Los Alamitos, USA, 1997, pp. 12–20.

[Sik95] T. Sikora, *Low complexity shape-adaptive DCT for coding of arbitrarily shaped image segments*, Signal Processing: Image Communication, vol. 7, November 1995, pp. 381–395.

[WCP05] K. M. Wong, K. W. Cheung, and L. M. Po, *MIRROR: an interactive content based image retrieval system*, IEEE International Symposium on Circuits and Systems, vol. 2, May 2005, pp. 1541–1544.

[WLW00] J. Z. Wang, J. Li, and G. Wiederhold, *SIMPLIcity: Semantics-sensitive Integrated Matching for Picture LIbraries*, VISUAL '00: Proceedings of the 4th International Conference on Advances in Visual Information Systems, 2000, London, UK, pp. 360–371.

[ZC08] E. Zavesky and S. Chang, *CuZero: embracing the frontier of interactive visual search for informed users*, MIR '08: Proceeding of the 1st ACM international conference on Multimedia Information Retrieval, 2008, Vancouver, British Columbia, Canada, pp. 237–244.

[ZO99] D. Zier and J. Ohm, *Common datasets and queries in MPEG-7 color core experiments*, ISO/IEC/JTC1/WG11, 1999, Melbourne, Victoria, Australia.